“Evercycle” - A Virtual Reality Exercise Game to Accelerate Exercise Performance Improvement using the “Feedforward Effect”

Alexander Whaley

Bachelor of Science in Computer Science with Honours
The University of Bath
May 2017

“Evercycle” - A Virtual Reality Exercise Game to Accelerate Exercise Performance Improvement using the “Feedforward Effect”

Submitted by: Alexander Whaley

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Signed:

❑ **Retain for Sample** **Mark**

UNIVERSITY OF BATH **Department of Computer Science** **INDIVIDUAL COURSEWORK** Submission Cover Sheet **Please fill in both columns in BLOCK CAPITALS and post into the appropriate Coursework Submission Box.**	*for office use* **Date and time received** **This section will be retained by the Department Office as confirmation of hand-in**
How to present your work 1. Bind all pages of your assignment (including this submission sheet) so that all pages can be read by the marker without having to loosen or undo the binding. Ensure that the binding you use is secure. Missing pages cannot be marked. 2. If you are required to submit part of the work on a disk, place the disk in a sealed envelope and bind the envelope into the submission. You must keep a copy of your assignment and disk. The original is retained by the Department for scrutiny by External Examiners.	**Declaration** *I certify that I have read and understood the entry in the Department of Computer Science Student Handbook on Cheating and Plagiarism and that all material in this assignment is my own work, except where I have indicated with appropriate references. I agree that, in line with Regulation 15.3(e), if requested I will submit an electronic copy of this work for submission to a Plagiarism Detection Service for quality assurance purposes.*
FAMILY NAME	FAMILY NAME
GIVEN NAME	GIVEN NAME
UNIT CODE	UNIT CODE
UNIT TITLE	UNIT TITLE
DEADLINE TIME & DATE	DEADLINE TIME & DATE
COURSEWORK PART (if applicable)	COURSEWORK PART (if applicable)
SIGNATURE	SIGNATURE

Abstract

National obesity levels are rising, and sedentary lifestyles are becoming more common in industrialized nations. Individuals within this category find maintaining and improving at exercise difficult, and feel a lack of motivation to continue exercising.

Gamification has been shown to be an effective way of helping users improve in a variety of tasks including exercise. We identify the key motivators and skills that help users improve at exercise, and find corresponding gamification techniques to keep users motivated and accelerate the rate at which users improve at exercise. Similarly we consider the barriers to traditional exercise and how we can avoid them in a virtual reality exercise game.

We identify “ghost” recordings as a widely utilised way of gamifying self-competition, a key factor in helping users improve. We investigate not only the established effectiveness of ghosts with regards to user performance and immersion/enjoyment, but also identify the potential to extend the effectiveness of ghost implementations by utilizing the psychological ”Feedforward Effect”.

We proceed to investigate design considerations based on previous virtual reality exercise game implementations and related literature. We use this to design our own exercise game that meets the needs of people who struggle to maintain/improve at exercise.

We implement our own virtual reality exercise game using appropriate exercise and virtual reality hardware. Utilizing this game, we conduct an empirical study to determine whether or not the feedforward effect can significantly contribute to accelerating the rate at which users improve at exercise, without negatively impacting the user experience.

Contents

1 Introduction **1**

1.1 Problem Description . . . 1

1.2 Research Problem . . . 2

1.3 Proposed Solution . . . 2

1.4 Contributions . . . 2

1.5 Outline . . . 3

2 Literature Survey **4**

2.1 Gamification and Improvement . . . 4

2.1.1 What is Gamification and How is it Used to Help People Improve? . 4

2.1.2 Gamification Effectiveness and the Key To Effective Gamification . . 6

2.1.3 Gamification and Exercise . . . 7

2.2 Exercise Motivation . . . 8

2.2.1 Why do people find it hard to keep exercising? . . . 8

2.2.2 What motivates people to exercise and improve? . . . 9

2.2.3 Gamification of motivational factors . . . 9

2.2.4 The Feedforward Effect . . . 11

2.3 Virtual Reality Exercise Games . . . 13

2.3.1 Why a Virtual Reality Cycling Game? . . . 13

2.3.2 The Hardware . . . 14

2.3.3 Game Design Considerations . . . 15

2.3.4 Health and Safety . . . 17

2.4 The Exercise Protocol . . . 18

2.4.1 Protocols in Existing Virtual Reality Exercise Games 18

2.4.2 High Intensity Interval Training 19

3 Requirements **21**

3.1 Hardware 22

3.1.1 The exercise bike must be capable of interfacing fully with the game engine. 22

3.1.2 The Head Mounted Display must be capable of tracking users "leaning". 22

3.1.3 The Hardware must be safe to use, hygienic and comfortable. 22

3.1.4 We must use an additional separate display so that we can observe the users perspective at all times. 23

3.1.5 We must use some hardware to record Heart Rate. 23

3.2 Exercise Protocol 24

3.2.1 The exercise protocol must be accessible and scale to each individuals skill level. 24

3.2.2 The exercise protocol must deliver an exercise experience based on the Wingate High Intensity Interval Training protocol. 24

3.2.3 The exercise format should be able to be changed to suit future needs/applications. 24

3.3 Game Design 25

3.3.1 The game character must be controlled by the users real world actions. 25

3.3.2 The game should have suitably challenging short term and long term goals. 25

3.3.3 The game should must engaging game mechanics to ensure users are in a state of flow. 25

3.3.4 The game must have an immersive aesthetic with clear distinctions between the games intensity states. 26

3.3.5 The game must include gamification of the motivational factor - "Challenge" (or Self Competition), in the form of a ghost. 26

3.3.6 The game must include gamification of the "Feedforward Effect" to determine it's effectiveness in a virtual reality exercise game. 26

3.3.7 The player must be able to see the ghost or know how far away it is at all times. 26

3.3.8 The HMD interface must display all relevant exercise statistics to the user. 27
3.3.9 The game must be able to record independent variables reflecting user performance. 27
3.3.10 The game must be able to save and read data between sessions. . . . 27
3.3.11 The game must run at a minimum of 90 frames per second. 27
3.3.12 The game must avoid sensory disconnects. 27

4 Design 28

4.1 Hardware Design 28
4.1.1 Computer 29
4.1.2 Vive HMD 30
4.1.3 LCD Monitor 31
4.1.4 Exercise Bike 31
4.1.5 Terminal Output 32
4.1.6 HTC Vive Lighthouses 33
4.1.7 Heart Rate Tracker and Samsung Tablet 33
4.2 Exercise Protocol Design 33
4.2.1 The Standard Wingate Protocol 33
4.2.2 Our Wingate Protocol Variation 34
4.3 Game Design 36
4.3.1 Game Overview 36
4.3.2 Level Design 37
4.3.3 Short Term Gameplay Mechanic - Avoid the Trucks 38
4.3.4 Controls 40
4.3.5 The Player View and The User Interface 41
4.3.6 Game Aesthetic 43
4.3.7 Long Term Gameplay Mechanic - Race The Ghost 46
4.3.8 Incorporating the Feedforward Effect 47
4.3.9 The Menu System 49
4.4 Software Design 53

4.4.1 Software Choices . . . 53
4.4.2 Script Structure . . . 55

5 Implementation 57
5.1 Optimization . . . 57
5.1.1 Mesh Combining . . . 57
5.1.2 Mirrors Implementation . . . 62
5.2 Interfacing with The Exercise Bike and Longitudinal Player Control . . . 63
5.3 Interfacing with HMD and Lateral Player Control . . . 66
5.3.1 Data Saving/Loading . . . 67
5.4 The Ghost and The FeedForward Effect . . . 69
5.5 Track Generation and Recording . . . 72
5.6 Tracking the Exercise Protocol and Changing Intensity States . . . 74

6 Evaluation 77
6.1 Methodology . . . 77
6.1.1 Experimental Design . . . 77
6.1.2 Hypotheses . . . 83
6.1.3 The Participant Demographic . . . 87
6.2 Results . . . 88
6.2.1 Performance . . . 88
6.2.2 Flow and Intrinsic Motivation/Enjoyment . . . 91
6.2.3 Perceived Difference . . . 92
6.3 Discussion . . . 93
6.3.1 H1: Using The Ghost with the Feedforward Effect (increased resistance) will improve exercise performance. . . . 93
6.3.2 H2: Using the Ghost with the Feedforward Effect (increased resistance) will improve exercise performance by a greater amount than just the Ghost without The Feedforward Effect. . . . 94
6.3.3 H3: There will be no significant difference in flow and intrinsic motivation/enjoyment when using the Ghost with The Feedforward Effect(increased resistance) and when using it without The Feedforward Effect(increased resistance). . . . 94

6.3.4 H4:Most people will not notice the increased resistance, when using the Feedforward Effect. 95

6.3.5 Experiment Limitations 95

6.3.6 Participants Opinions on our game - "Evercycle" 96

7 Conclusions 97

7.1 Contribution 97

7.1.1 A Comprehensive Evaluation of the Needs of People that Struggle to Maintain/ Improve at Exercise. 97

7.1.2 Identified Suitable Game Mechanics that can Help People Improve at Exercise. 97

7.1.3 Effective Gamification of Challenge/Self-Competition via a Ghost. . 98

7.1.4 Extended the Functionality of Ghosts using the Feedforward Effect. 98

7.1.5 Conducted an Empirical Study To Evaluate the Effectiveness of our Feedforward Ghost 98

7.2 Future Work 98

7.3 Reflection 99

A University Health Screening Questionnaire 105

B Participant Instruction Sheet 109

C Participant Consent Form 116

D Ethics Checklist 119

E Questionnaires 122

F Raw Data 123

List of Figures

2.1 Scientific Literature Search Hits for Gamification (Hamari, Koivisto and Sarsa, 2014) . . . 5

2.2 Five Step Process to Gamifying a Concept (Huang and Soman, 2013) . . . 5

2.3 "Pokemon Go" In Use . . . 8

2.4 Statistical Analysis of Exercise Motives Across Exercise Stages (Ingledew, Markland and Medley, 1998) . . . 10

2.5 "Mario Kart" using Ghosts of Previous Performance for Self Competition . 11

2.6 Successful Studies making Use of the Feedforward Effect (Dowrick, 2012*b*) . 12

2.7 Mean Enjoyment Scores (On a 1 - 7 Scale) (Shaw, 2014) . . . 14

2.8 Mean Motivation Scores (On a 1 - 7 Scale) (Shaw, 2014) . . . 14

2.9 The Hardware Setup Used in an Existing Exercise Game (Shaw, Wnsche, Lutteroth, Marks and Callies, 2015) . . . 15

2.10 A Flow State Visualization . . . 16

2.11 Participants time till exhaustion before (PRE) and after (POST) six sessions of HIIT over a two week period vs Participants with No Training Intervention (Gibala and McGee, 2008) . . . 19

2.12 Participants $\mathbf{VO_2}$ **Max** before (PRE) and after (POST) six weeks of Wingate-based HIIT over a two week period vs Participants undergoing 6 weeks of Traditional Endurance Training (Gibala, Little, MacDonald and Hawley, 2012) 20

4.1 An Overview of the Hardware Setup we are using for our Virtual Reality Exercise Game . . . 28

4.2 The Practical Hardware Setup in the University DASH Lab . . . 29

4.3 HTC Vive Axis Tracking . . . 30

4.4 HTC Vive Cable Bundle Body Clearance . . . 31

4.5 Terminal Output Currently Displaying Resistive Force T and Power P . . . 32
4.6 The Position of Lighthouse 1 32
4.7 The Position of Lighthouse 2 32
4.8 Lode's standard Wingate Protocol Breaking Torque Values for Sections of High Intensity. [1]. 34
4.9 Flow Diagram of Our First Exercise Protocol Design 34
4.10 Flow Diagram of Our Final exercise Protocol Design 36
4.11 Our Three Lane "Infinite Runner Style" Level Design 37
4.12 Screenshots from the hit Mobile Game - "Subway Surfers" 37
4.13 A Randomly Generated Track with Fixed Tile Size 38
4.14 Some of the Different Tiles Used by the Procedural Track Generator 39
4.15 Trucks the User Has to Avoid Whilst Playing the Game 40
4.16 The Wing Mirrors Fit to the Bike Model to Allow Users to See Backwards . 42
4.17 The In-Game User Interface 42
4.18 The Game Aesthetic in the Low Intensity Phases 44
4.19 The Game Aesthetic in the High Intensity Phases 45
4.20 The Distance Police Cars are from the Player at the Start of the High Intensity Sprint 45
4.21 The Distance Police Cars are from the Player by the End of the High Intensity Sprint 45
4.22 The "Ghost" as Seen In Game by the Player. 46
4.23 Screen Fade to Black Effect, Before Teleporting the Player 46
4.24 A Still Image from the Short Introduction Sequence That Helps The User Identify As The Game Character 48
4.25 The Menu System State Diagram 49
4.26 The Main Menu 50
4.27 The Ready Screen 51
4.28 The Finish Screen 52
4.29 The Results Screen 53
4.30 The Script Structure Diagram 55

5.1 Non-Optimized Environment Batch Count 58

5.2 Optimized Environment Batch Count 62
5.3 HTC Vive Axis Tracking 66

6.1 A Graph Illustrating the Average Total Energy Used in Each Condition . . 89

List of Tables

6.1 Performance - The Total Power used By Each User across Each Condition . 89

6.2 Performance - The Performance Increase of Condition 2 and 3 in Comparison with respect to Condition 1 90

6.3 Average Flow Values Across all 3 Conditions 91

6.4 Enjoyment Levels Across all 3 Conditions 92

Acknowledgements

I'd like to give a heartfelt thanks to Dr Christof Lutteroth, not only for his guidance throughout the project, but also for his unbridled enthusiasm in my work. He truly made the project feel like something that mattered and I in turn cared about game I was creating all the more because of it.

I'd like to extend my gratitude to both Soumya Barathi and Daniel Finnegan without whom, my game would never have even communicated with the exercise bike.

I'd like to send similar thanks to Christof's team in Auckland who provided an array of advice regarding questionnaires and the experiments methodology.

Also a big thanks, to my friends and family who encouraged me all the way to the very end.

And one final enormous thank you to Jake Allan for assisting me in proof reading, so close to the exam period.

Chapter 1

Introduction

In this chapter we highlight the problem description and building on this highlight the specific research problems surround it. We propose a suitable solution, define our contributions and then discuss the outline of paper.

1.1 Problem Description

Rates of obesity are increasing nationality, 27% of adults in England are obese and a further 36% are overweight (Baker, 2017). Obesity increases the risk of other health issues dramatically such as: joint problems, coronary heart disease, hypertension and more. These figures are increasing and continue to increase because sedentary lifestyles have become more common place in industrialized nations. In Scotland the average man is sedentary for 5.5 hours a day and women are sedentary for 5.4 hours a day. Sedentary life styles increase the risk of heart and circulatory disease.

These groups of individuals find maintaining and improving at physical exercise difficult (Dishman, 1991). These groups of people are considered non-maintainers of exercise. These non-maintainers feel a lack of motivation to continue exercising (Cropley, Ayers and Nokes, 2003), a study found that almost50% of overweight people felt like they couldn't exercise was because they were "Not Motivated/Couldn't Get Started".

Gamification is an emerging concept with by which developers use game-design elements in non-game contexts, with the aim to motivate and increase user activity and retention. Numerous studies have shown gamification to be a a viable technique to help motivate users to exercise. Virtual reality exercise games in particular are an emerging area, with only a small literature following, but big potential to help users improve, through increased immersion.

1.2 Research Problem

Exercise games show great potential in helping users get motivated and improve at exercise. However, there are no design standards in place when it comes to creating a virtual reality exercise game, and in order for gamification to work successfully developers need to ensure that the design decisions they make meet the need of their users (Huang and Soman, 2013).

What exactly are the needs of people who struggle to main/improve at exercise, and what are the skills they are lacking or the barriers to traditional physical activity that stops these users from maintaining exercise? Which of these skills/factors can best be gamified to help users improve at exercise, what design considerations need to be accounted for when designing a virtual reality exercise game?

Are there any techniques that can be deployed in a virtual reality exercise game to speed up the rate at which users can improve/increase their exercise performance?

1.3 Proposed Solution

We intend to identify the exact needs of users who struggle to maintain and improve at exercise. Using this we will identify the key skills they are missing with regards to exercise improvement/motivation, we will attempt to offload these skills into a virtual reality game so that these users can maintain exercise with being required to have the normal traits found in people who can maintain/improve at exercise.

We will determine a definitive list of game design decisions based on that must be adhered to create a successful virtual reality exercise game. We will create our own implementation of a virtual reality exercise game that implements game mechanics that attempt to help users improve their exercise performance at an accelerated rate.

1.4 Contributions

- We will definitively identify and document what skills people who struggle to main and improve at physical exercise are lacking.
- We will build off existing gamification techniques and design recommendations to create out own virtual reality exercise game.
- We will incorporate a “ghost system” into our game, where users will race against recordings of themselves, to improve between sessions.
- We will incorporate the ”Feedforward Effect” to some degree in an attempt to further accelerate user performance improvement.
- We will conduct a study that determines the effectiveness of this effect in a virtual reality exergame in conjunction with a ghost.

1.5 Outline

We start by doing an extensive literature review into the effectiveness of the history and effectiveness of gamifcation to determine what's required for successful gamification of a task. We then investigate the exercise motivation to identify what motivates people to exercise and the way we can effectively gamify these skills/factors. We study existing virtual reality exercise games and specifically those that implement a ghost, we research the Feedforward effect to determine it's nature and applicability to this project. Finally we determine the most appropriate exercise protocol to use.

Using the information we uncover in our literature review we establish some large granularity requirements for the hardware, the exercise protocol and finally the game design. These requirements are assigned priorities with which they must be incorporated with respect to.

To fulfill the requirements we structure a design for our game. This design is broken down into four main components that coincide with the requirements breakdown. We create a hardware system design, an exercise protocol design, a game design and software structure design. When determining our design decisions we consider multiple options and validate our choices based on established requirements and/or other literature. This established design is then implemented, in whichever way we determine appropriate.

We continue on to discuss the nuances that occurred in the implementation of our design and potentially how implementation details ended up dictating parts of the design. We discuss in great detail the problem of optimization which was not covered in design but was of paramount importance to the successful delivery of the final product.

With a working implementation we subsequently discuss the intentions of our experiment and the dependent variable we intend on measuring when varying the ghost and the implementation of the Feedforward Effect. We detail the the entire experiment procedure - which was incredibly thorough in order to be approved by the Universities Health Research and Ethics Committee. We then evaluate the results of our aforementioned study with respect to hypotheses we state that will relate to the use of the Feedforward Effect, and it's effects on performance and the user experience.

Finally we draw a conclusion to our project and discuss the main contributions our project has made to it's related fields. We go on to discuss potential future work that makes use of the project and finally give some personal reflection into the entire process.

Chapter 2

Literature Survey

In this chapter, we investigate current literature surrounding the effectiveness of gamification. We look at how gamification techniques can be applied to exercise motivators and demotivators in an effort to create a game that meets the needs of people who struggle to maintain/improve at exercise. We look at the Feedforward Effect and consider its validity and its integration into our virtual reality game. Furthermore, we determine key design considerations of virtual reality exercise games based on existing literature and other virtual reality exergames to create the most effective and accessible game possible. We finish by determining a suitable exercise regime that best fits the nuances of a virtual reality exercise game system.

2.1 Gamification and Improvement

2.1.1 What is Gamification and How is it Used to Help People Improve?

The first documented use of of the term "gamification" dates back to 2008. However, the term did not see widespread adoption until 2010 (Deterding, Dixon, Khaled and Nacke, 2011). It has seen a rise in scientific critique since 2010 (See Figure 2.1), however, until recently it was somewhat of an undefined concept and an industry umbrella term, meaning its legitimacy has often been challenged (Raczkowski, 2013). Recently gamification has been defined formally as: "The use of game-design elements in non-game contexts, with the aim to motivate and increase user activity and retention." (Deterding et al., 2011).

The key concept behind gamifcation is that if we can identify and isolate the active components of games that makes them fun and addictive, we can apply these components to non-game activities to make them similarly fun and engaging. The issue with this however, is that there is non standardized list of these "game elements", as often developers disagree on what components are most significantly contributing to enjoyment/retention of games (Cugelman, 2013). Despite this, here is a shortlist of commonly used game mechanics

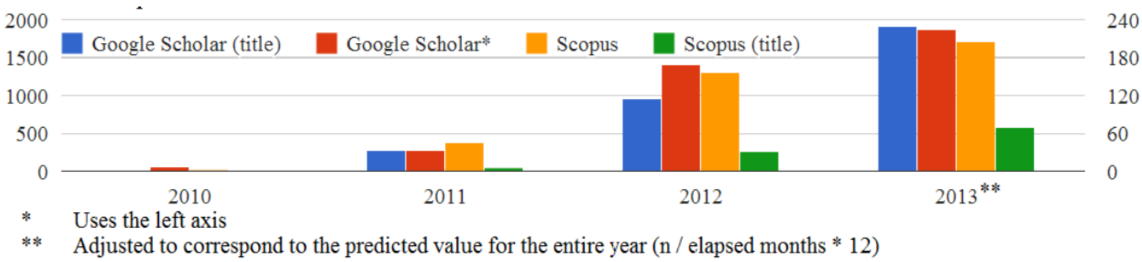


Figure 2.1: Scientific Literature Search Hits for Gamification (Hamari et al., 2014)

present in the gamification of non-game activities:

- Achievements (Experience points, Levels, Bonuses etc.)
- Exercises (Challenges, Discoveries etc.);
- Synchronizing with the Community (Leaderboards, Collaboration etc.);
- Result Transparency (Experience bars, Continuous feedback etc.);
- Time (Countdown, Speed etc.);
- Luck (Lottery, Random Achievements etc.).

The problem with defining very clear cut game design mechanisms that will make gamification effective is that the elements that are most effective will be dependent on a few very key factors. Namely the target audience, the objective of the task and the resources available. When gamifying a task/concept, these factors can be systematically considered using the following five step process (See Figure 2.2) (Huang and Soman, 2013).

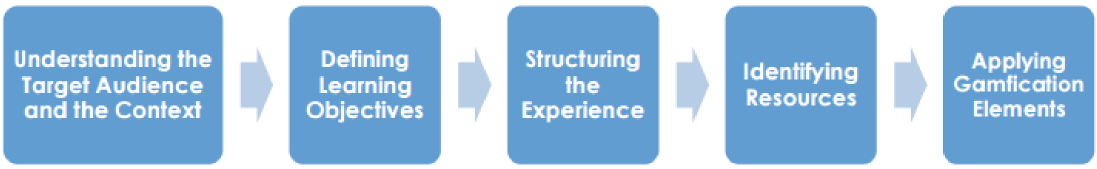


Figure 2.2: Five Step Process to Gamifying a Concept (Huang and Soman, 2013)

Gamification is used in an assortment of professional fields with the common goal of improving the user experience and performance. It has seen widespread successful applied use and seen extensive literature review in education (Huang and Soman, 2013; Charles, Charles, McNeill, Bustard and Black, 2011; Burguillo, 2010) , business (Gears and Braun, 2013; Kumar, 2013) and health/exercise (Cugelman, 2013; Yim and Graham, 2007; LeBlanc and Chaput, 2016; Lister, West, Cannon, Sax and Brodegard, 2014).

2.1.2 Gamification Effectiveness and the Key To Effective Gamification

It is now important to consider whether studies into the effectiveness of gamification deem it a successful technique in helping people improve, such that we can confidently create a game with the knowledge that it has the potential to help users improve and make the task more enjoyable.

Broadly across all fields, a very extensive literature review was conducted in 2014 (Hamari et al., 2014). This literature review consisted of 40 different papers that were investigating the effectiveness of gamification with regards to motivational affordances, psychological outcomes and further behavioral outcomes within users. The paper simply concludes that with regards to motivating users, improving their psychological state during the activity and users' future outlook on the tasks, gamification does indeed work.

However, we should consider more specific research into the effectiveness of gamification to see how and why gamification is successful. It is also important to consider fields other than just health/exercise, as there may be key factors regarding successfully gamifying tasks/activities that similarly cross over to the health/exercise gamification format.

Education, a 2013 piece of literature by Huang, investigated an assortment of real world case studies (Huang and Soman, 2013). It investigated the gamification of tasks such as problem sheets/homework hand-ins, content learning and simple class work. The different case studies used various techniques including but not limited to: work theming, experience tracking based off work completion and competition based leader boards amongst students. Case study instances found that: 76% of students found the system helpful in their learning and 71% felt that experience and leveling up helped motivate them to finish assignments. The overall conclusion of the paper found that in comparison to a traditional working environment, where student motivation to learn can be hindered by an array of factors, appropriate application of gamification techniques lead to information being delivered in a way that students found more enjoyable and addictive. Gamification effectively assists in reducing negative emotions encountered in traditional forms of education, and allows students to make use of the "learn by failure" technique, that is popular in game-like environments (Huang and Soman, 2013).

Feedback for tasks in education is critical to empowering a user to establish and understand their educational identity. A game-based feedback approach conducted in 2011 harnessed unique aspects of virtual worlds to enhance user feedback and engagement (Charles et al., 2011). The use of personal avatars within this game allowed users to further engage/relate to their task and make more informed choices within their learning context (Charles et al., 2011). The use of competition based gamification, both self competition (competing against your previous scores) and against other users scores, was also found to be an extremely effective gamification technique in Education related tasks, keeping users motivated as well improving their student performance(Burguillo, 2010).

It is similarly important to investigate the effectiveness of gamification in business, especially since we should consider how effective gamification is for older generations and in a more professional environment. A paper investigating gamification techniques in a business environment claims gamification is 75% psychology and 25% technology, recommending that the start to selecting suitable and successful game mechanics is to first understand the psychology behind the task (Kumar, 2013). By understanding what motivates/demotivates people when carrying out a task, we can select game mechanics to help reduce the negatives and add positives to the experience. Extending upon this, Dale states that it has been predicted that whilst the use of gamification is rapidly increasing to near multi-billion investment, around 80 percent of future gamified applications will fail to meet business objectives. Studies believe that the key to successful gamification is about connecting to people on an emotional level and motivating them to achieve their goals (Dale, 2014). Some studies believe that designers will often neglect to consider the potential user when gamifying a process which can lead to noneffective gamification. On the other hand, gamification yields positive results when the implementer focuses on who their target audience is, the behaviour they want to improve/change and the motivating factors that lead to engagement and improvement (Kumar, 2013; Gears and Braun, 2013).

2.1.3 Gamification and Exercise

The effectiveness of gamification in health/exercise has been extensively reviewed and is very relevant to coming to an effective solution to the research problem. Currently, there is an abundant use of gamification in health and fitness apps and 60 percent of health initiatives in work spaces now include gamification elements (Lister et al., 2014). Reviews investigating the effectiveness of gamification in health and exercise applications believe that gamification is an effective way to increase motivation in users. However, in a largely over-saturated application market, current industry use of gamification in relation to behavioral theory is sub-par. This may be in part due to a lack of a well defined industry standard (Kumar, 2013).

There are strong ties between gamification techniques and and behavioral science principles that have been proven to work in digital health behaviour change interventions (Cugelman, 2013). This provides links to scientifically proven improvement methodologies in an exercise format, whilst also having the added benefit of having fun/playful qualities that can help with task engagement. Thorough review of the design requirements an exercise game should adhere to ensure players are always motivated (Yim and Graham, 2007). It was determined that when designing motivational exercise games, designers should be mindful of problems regarding poor exercise self-identity and low self-efficacy.

Study into the effectiveness of interactive video game exercise found that video game exercise is effective in enhancing exercise adherence. Considering BMI data, their findings imply its use has contributing factors towards the prevention of obesity (Warburton, Bredin, Horita, Zbogar, Scott, Esch and Rhodes, 2007). Similarly, research has indicated that the immersion provided by a virtual environment also assists in motivating users.

In 2016, the global hit augmented reality mobile game "Pokemon GO" (See Figure 2.3) effectively gamified exercise for millions of people. The recent of success of Pokemon GO

Figure 2.3: "Pokemon Go" In Use

should provide a large amount of data, regarding the effect it had on the levels of physical activity its users participated in. Moreover, there should potentially be significant amounts of qualitative feedback regarding the user experience when exercising using this game. Since the release of Pokemon GO, objectively measured statistics from Apple Health, as well as subjective measures (i.e. in game success and user reports) have suggested that playing Pokemon GO may be an effective strategy to increase time spent walking and increasing daily step counts (LeBlanc and Chaput, 2016). In addition, people are spending more time when playing this game, participating in active social activity as opposed to the sedentary activities they may usually be engaged in.

2.2 Exercise Motivation

2.2.1 Why do people find it hard to keep exercising?

It is important that we first understand the psychology behind why people find maintaining/improving their exercise performance difficult. Only then can we come up with appropriate gamification techniques to make the task easier and more engaging (Kumar, 2013).

Self-motivation stems from self-regulatory skills such as effective goal-setting, self-monitoring of progress, and self reinforcement (Dishman, 1991). People who struggle to maintain and improve at exercise often lack these self-regulatory psychological traits, however studies have been done that show these skills can be offloaded into game mechanics to help motivate users (Shaw, Buckley, Corballis, Lutteroth and Wuensche, 2016).

The main factor as to why people cannot maintain/improve at exercise is motivation. A study in 2008 by the School of Health in Victoria found that the most prominent reason overweight people felt they couldn't exercise was because they were "Not Motivated/-

Couldn't Get Started", with almost 50% of people agreeing with this statement (Ball, Crawford and Owen, 2000). By identifying the key motivational factors that help people sustain and improve at exercise, we can find corresponding gamification techniques to increase user motivation.

2.2.2 What motivates people to exercise and improve?

As well as considering the negative reasons as to why people do not exercise, we should also consider the reasons people do exercise to see if we can gamify any of these motivational factors.

The most immediate determinant of human behaviour is motivation. To facilitate permanent exercise motivation for exercise, it is critical that the users enduring goal is based on autonomy and should lead to feelings of enjoyment and competence (Iso-Ahola and Clair, 2000). If the exergame we create can provide this overall goal, then it should be able to help motivate it's users to exercise and improve.

A study into the correlation between progression through exercise stages of change and peoples exercise motivators found a statistically significant correlation between a few specific exercise motives and people who maintained their exercise. (Ingledew et al., 1998). With the goal of helping people improve and continue exercising in the future, we are most concerned with analysis that explains the transitions from the baseline study exercise stage to the follow up studies exercise stage. The factors that contributed the most to helping people to progress to, or remain at, a stage of maintained exercise were:

- Enjoyment
- Revitalization
- Competition
- Challenge

See Figure 2.4 for results of the statistical analysis of this data.

2.2.3 Gamification of motivational factors

Studies into the effectiveness of certain gamification techniques have found that often the self-regulatory psychological traits that people who cannot maintain/improve at exercise lack can be offloaded to game mechanics such as personal trainers, or pre-determined game goals and can thusly increase user motivation (Shaw et al., 2016). The use of virtual trainers (i.e. on screen prompts or audio queues) have been shown to be a viable way of helping motivate users and lower tension surrounding performance (IJsselsteijn, Kort, Westerink, Jager and Bonants, 2006). On screen data is useful to offload important motivational traits such as effective goal-setting, self-monitoring of progress, and self reinforcement, to the game rather than be a requirement of the user (IJsselsteijn et al., 2006). This means even users lacking key skills involved in exercise motivation can exercise more intensively and report higher intrinsic motivation (Cornick and Blascovich, 2015).

Table 5. **Discriminant analysis using motivations[a] to explain transitions from baseline to follow-up**

Discriminating variable	Correlation between discriminating variable and discriminant function
Affiliation	.33
Appearance	.21
Challenge	.47
Competition	.57
Enjoyment	.76
Health Pressures	−.44
Ill-health Avoidance	.14
Nimbleness	.30
Positive Health	.42
Revitalization	.69
Social Recognition	.40
Strength and Endurance	.45
Stress Management	.33
Weight Management	.04
Transition	**Value of discriminant function at group centroid**
Stayed inactive (*n* = 71)	−0.97
Stayed active (*n* = 126)	0.59
Became active (*n* = 26)	0.14
Became inactive (*n* = 23)	−0.38

Figure 2.4: Statistical Analysis of Exercise Motives Across Exercise Stages (Ingledew et al., 1998)

Now we consider the 4 most important exercise motives as detailed earlier (See 2.2.2), the first of which is enjoyment. Increasing enjoyment is the goal of gamification. By adding game elements to a non-game activity, it should make improve user performance and make the task more enjoyable (Deterding et al., 2011). Selecting appropriate gamification techniques that meet the other needs of users should also make the exercise activity more enjoyable. Effective gamification has been shown to quantifiably increase enjoyment (Herzig, Strahringer and Ameling, 2012). Revitalization is quite an abstract property and will be quite hard to tangibly find gamification mechanics to provide it to users, so we will not focus on this for our exercise game.

Conversely, competition and challenge translate incredibly well to a game environment. Competition is large part of what makes a wide array of games enjoyable, and has seen widespread use as a gamification technique in other health and other fields (Shaw et al., 2016; Burguillo, 2010). The difference between competition and challenge is who the user is competing with; in competition, users compete with other users whereas challenge refers to self competition. In a competition environment, there is an alienating issue regarding non-competitive individuals. These individuals experienced reduced enjoyment and motivation. Moreover, they performed worse when exercising in a competitive environment (against other players)(Song, Kim, Tenzek and Lee, 2010).

Conversely, studies show that challenge (self-competition), in this instance through racing a “ghost” of your previous performance (See Figure 2.5 for an example of a ghost in globally popular game “Mario Kart”), was better received by users when considering their enjoyment and performance (Shaw et al., 2016). There has not been particularly extensive

Figure 2.5: “Mario Kart” using Ghosts of Previous Performance for Self Competition

study into the effectiveness of ghost recording of previous performances in virtual reality exercise games. However, one study found that when users competed with their ghost on average their total calories burnt increased (Shaw et al., 2016). However, users reported only feeling motivated by the ghost when it was in front of them as they could not see it easily when it was behind them.

This technique of gamifying individual self competition or challenge to motivate and increase enjoyment for users will become the backbone of the exercise game we are creating. We will also build upon the idea of racing against yourself by incorporating the Feedforward Effect, with the aim of accelerating the rate at which users improve.

2.2.4 The Feedforward Effect

Feedforward is sometimes referred to as “feedforward learning” or even“self-modeling” but for the duration of this paper we will refer to it as feedforward or the feedforward effect. Feedforward makes use of the psychological concept of “future imaging”. By visualizing themselves achieving a certain goal, users are more motivated and instinctively learn the behaviours required to achieve this goal (Dowrick, 2012*b*). The most relevant example that we will be most closely relating to in our game is that of “Video Self-Modeling” in which users watch videos of themselves completing tasks they previously have been unable to complete (often through 3d-modeling or camera tricks). This in turn often and somewhat unexplainably assists users in achieving these goals. (Dowrick, 2012*a*).

The feed forward effect has not been widely utilised in exercise games or even video games in general. However, its inherent nature should lend itself well to the idea of helping users improve in a virtual reality exercise game setting. There have been extensive studies regarding the efficiency with which Video Self-Modeling works in the fields of exercise, education and communication.

A study investigating the effectiveness of VSM in the rehabilitation of Cerebral Palsy patients smiles was carried out in 2006. The patients were shown videos of themselves smiling for one minute, three times a day for a total of ten days. The study claims that post-treatment, a large amount of patients regained a “normal” smile and their smile reaction times increased by an average of 224ms. (Coulson, Adams, O’Dwyer and Croxson, 2006).

In the field of exercise, a study was conducted in which participants were asked to run until exhaustion. In one condition, participants were shown a video of themselves running at their optimum stride/pacing. In the other condition, the other participants were given a blank white screen to look at. The study found that the use of Video Self-Modeling to achieve the perfect pace assisted users in running for an average time of 19% longer than the control condition (Hagin, Gonzales and Groslambert, 2015).

There is significant scientific evidence to suggest the the Feedforward Effect and specifically Video Self-Modeling works to some degree. A study by Dowrick et al. investigating the effectiveness of The Feedforward Effect and specifically Video Self-Modeling compiled this table of studies in which VSM was used to some significant success (See Figure 2.6).

Reference; Participant(s)	Condition; Prior Treatment or Comparison	VSM “Dose”	Outcomes
Dowrick & Raeburn, 1995; 5-year-old girl with cerebral palsy	Could not step over 1-cm obstacles on smooth floor; PT & OT for 6 months with no progress	12 min (2 min x 6, over 2 weeks)	Stepped over 6-cm obstacles plus outdoor curbs
Kahn et al., 1990; adolescents with mild-to-moderate depression	Primary group comparison: 30 hr of CBT in 8 weeks	21 min spread over the same 8 weeks as the comparison interventions	Significant improvements, standard measures; 21 min VSM = 30 hr CBT
Kehle et al., 1990; 7-year-old boy, selective mutism	Case study, subsequent to 3 years of not speaking at school	3 min (viewed once)	Spoke freely, introduced all classmates to visitor next day
Buggey, 1995; three 5-year-olds with language delay	Multiple baseline; use of verb *to be* 0%–15%	2 min (first viewing)	Use of verb *to be* 30%–60%; generalization
Coulson et al., 2006; 10 adults with awkward, asymmetric smiling from facial nerve damage	10 case studies, up to 8 years of standard rehabilitation with no progress	30 min (1 min, 3 × per day, 10 days)	Friendly smiles, changed lives; reaction times, 224 ms faster
Murphy & Davis, 2005; deaf children	Case studies; previous skill level low	3 min (first viewing)	Signing words; number and use tripled
Bellini et al., 2007; children with autism, 4–5 years old	Multiple baseline; baseline participation, 4.5%	20 min (2 min × 10)	Unprompted social engagement in conversation increased 42%

Note. PT = physical therapy; OT = occupational therapy. Adapted from Dowrick (in press) with permission of the publisher, John Wiley & Sons, Inc.

Figure 2.6: Successful Studies making Use of the Feedforward Effect (Dowrick, 2012*b*)

We will consider when designing our exercise game a way in which the feedforward effect can be modified and thusly integrated into our system in conjunction with the users previous "ghost" recordings, to potentially accelerate the rate of user improvement in a short time.

2.3 Virtual Reality Exercise Games

2.3.1 Why a Virtual Reality Cycling Game?

Virtual Reality provides an immersive experience and, as previously discussed, increasing immersion increases motivation (IJsselsteijn et al., 2006; Li, Maxwell, Leightley, Lindsay, Johnson and Ruck, 2014). When exercising using a virtual reality exercise game, studies have shown that users achieve a degree of attentional distraction from bodily sensations. This means that users can potentially exert themselves further without feeling extra muscular duress (Banos, Escobar, Cebolla, Guixeres, Alvarez Pitti, Lisón and Botella, 2016). Disassociation from exercise through the use of emotive virtual worlds should aid with user performance. A study into the effects of exercise disassociation have shown that athletes report lower perceived exertion when dissociated from the exercise in such a way (Masters and Ogles, 1998; Morgan and Pollock, 1977).

There have been a few virtual reality exercise games created in recent years that have made use of an exercise bike, which most developers consider the best option as it allows users to remain seated and reduces the risk of injury or motion sickness (Cornick and Blascovich, 2015). Scientific studies were conducted on some of these games which gives us a good basis to justify this as an appropriate format for our virtual reality exercise game.

One game called "PaperDude" sees users cycling down a road using hand gestures to throw out newspapers to score points. The scientific review believes the game provides an immersive exercise experience with, they state that the use of a natural form of game input (body controlled action) encourages exercise, as a byproduct of playing the game (Bolton, Lambert, Lirette and Unsworth, 2014). It also raises some interesting design issues that we will draw on shortly.

A study done by the University of Auckland measured and recorded the mean ratings for both enjoyment and motivation of users playing an exercise game (Shaw, 2014). In this study they had users carry out the traditional cycling exercise, play the game on a monitor display, and finally play the game with a virtual reality headset. The study found that user enjoyment and motivation both increased dramatically across both conditions in which the game was used within all user subgroups. Building on this, enjoyment increased further across the board when users played the game using the virtual reality headset, and motivation increased by near-similar amounts (See Figures 2.7 and 2.8).

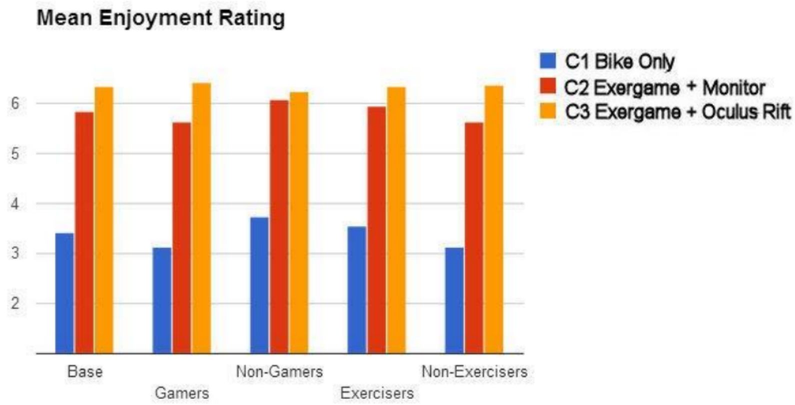


Figure 2.7: Mean Enjoyment Scores (On a 1 - 7 Scale) (Shaw, 2014)

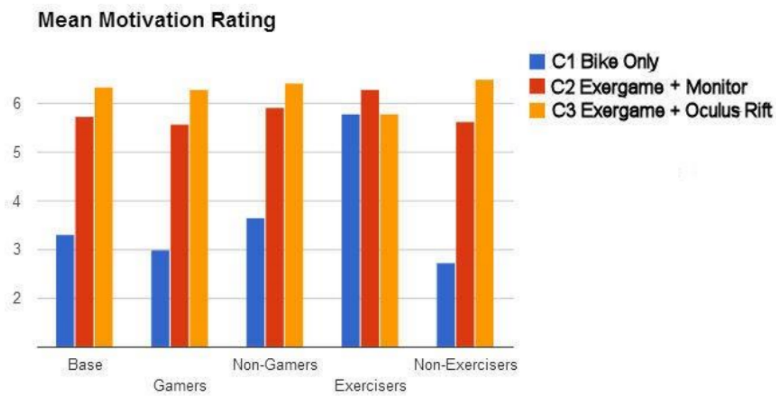


Figure 2.8: Mean Motivation Scores (On a 1 - 7 Scale) (Shaw, 2014)

2.3.2 The Hardware

Existing virtual reality exercise cycling games in the past have all used similar set-ups. Firstly using an exercise bike that in some way interfaces with the exercise game, a virtual reality headset (in these examples the Oculus Rift since this was before the HTC Vive was released) and a Kinect to track the position of the rider.

Shaw et al. used the following set-up (See Figure 2.9) and had the following points to consider regarding their hardware choices. They determined that the bike they were using would need the ability to vary resistance and display the RPM. Moreover, this needed to be done in such in a way that interfaces seamlessly between the Unity game engine they were using. In their system, they unfortunately did not have access to a bike capable of this (Shaw et al., 2015) but we should aim to.

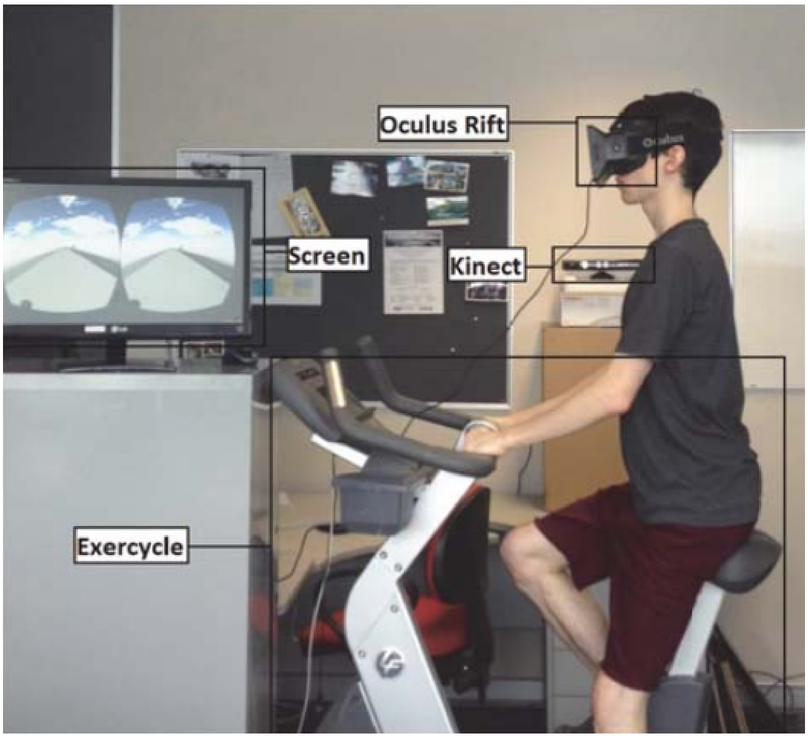


Figure 2.9: The Hardware Setup Used in an Existing Exercise Game (Shaw et al., 2015)

They used an Oculus Rift as their virtual reality headset and a Kinect to track the users body position to effectively control the in game character. Using these two pieces of hardware caused some issues as the Kinect runs at exactly 30 frames per second which is poorly compatible with virtual reality due to anything less than 90 frames per second causing motion sickness (Kolasinski, 1995). The designer of "PaperDude" also found that lean-based control systems do not work well with the Kinect due to limitations in it's sensing. This lead to users having to lean too far in either direction and proposed too significant a safety risk (Bolton et al., 2014). For these reasons we believe it is best to avoid using the traditional Kinect system and instead find an alternative hardware solution.

2.3.3 Game Design Considerations

Drawing from papers on virtual reality exercise game design considerations as well as the feedback from existing virtual reality exercise games, we identify key areas of game design that we will need to consider in order to make a successful virtual reality exercise game.

Aesthetics - the theming and art style of the game will define the context in which the game takes place. Game aesthetic renders an atmosphere that draws and maintains players attention and immersion on an emotive level (Fabricatore, 2007). Considering this, we should try to make the aesthetic of the game match the current experience/mindset we want to evoke in our users.

Gameplay - The actual game tasks the user must deal with should be considered carefully; if the game doesn't offer enough mechanical gameplay, then the user will just be focused on the exercise which defeats the purpose of using an exercise game. Conversely, if the tasks are too involved or distracting, then the user wont be able to exercise to their fullest. The main mechanic of "PaperDude" involved users throwing papers meaning they'd have to take their hands off the bike which would significantly reduce the potential exercise performance and is potentially a safety risk. (Bolton et al., 2014). Ideally we want people to be in a state of "Flow". Flow is described as being a psychological state in which users are utterly immersed in achieving their goal/task with little consideration of environment outside the task. In order to achieve this state, the task difficulty must be perfectly balanced; not too hard to cause frustration and not too easy as to cause boredom (See Figure 2.10 for a visual representation of the flow state). (Csikszentmihalyi, 2014).

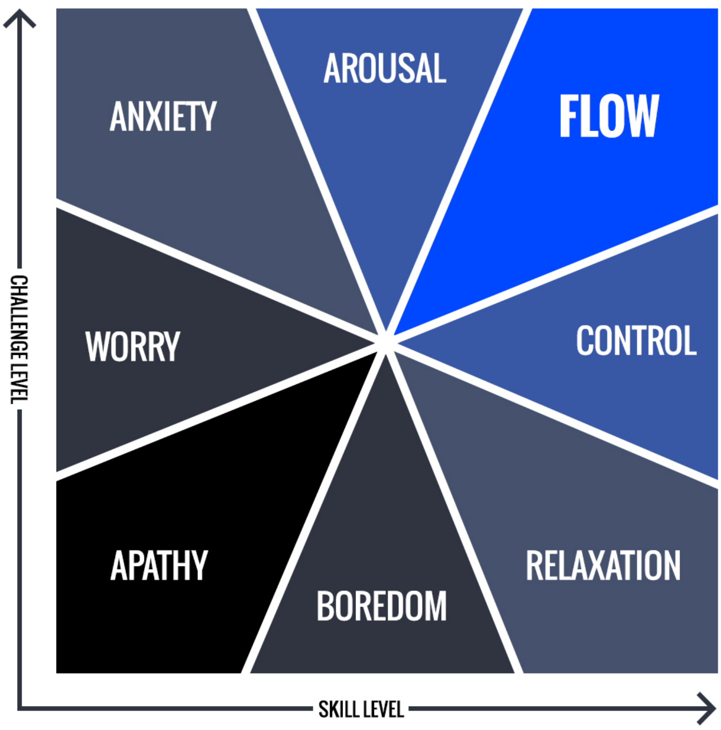


Figure 2.10: A Flow State Visualization

User Interface - As previously mentioned, by offloading useful, motivational skills such as goal-setting, self-monitoring of progress and self reinforcement means users are not required to have these traits to remain motivated (IJsselsteijn et al., 2006). If we offload information such as time left in sections, the users current speed, motivational on screen prompts and also define the exercise goal the user should be achieving, then even people without these key motivational skills can can exercise more intensively have higher intrinsic motivation (Cornick and Blascovich, 2015).

Control Scheme - The control system should be intuitive, in a virtual reality system usability is greatly improved by making sure your in game actions are well represented by corresponding real world actions. For example, to throw newspapers in the game "Paper-Dude" the user simply throws with their arm in real life. (Bolton et al., 2014). In the exercise game made by Shaw et al., they first mapped users leaning to turn the character in game left and right. Whilst this was okay on their monitor display, significant motion sickness was introduced with the Head Mounted Display. (Shaw et al., 2015). They instead opted for leaning to translate to horizontal movement across a straight path; a control scheme we will likely build upon. It is very important that within our game there are no sensory disconnects, if users actions in the real world do not translate in a way that coincides with the respective real world action this induces significant nausea.

Scalability/Accessibility - Starting an exercise game can be intimidating to new players so the system should scale to the performance of the individual user and the system should avoid systematic barriers between skill levels (Yim and Graham, 2007). By introducing a difficulty scale, users will not be discouraged by a lack of physical ability. This also ties into a similar consideration raised by Yim et al.; it is important that fitness levels of users should be hidden as to not demotivate them. If we make the game experience scale such that everyones experience is consistent regardless of the relative physical effort exerted, then users won't feel like they're not good enough to use the system (Yim and Graham, 2007).

Goals - We've previously discussed that goal setting is a key skill in maintaining user motivation (See 2.2.1). Short and long term goals within the game are important to establish to maintain exercise motivation within sessions and also between sessions (Yim and Graham, 2007).

2.3.4 Health and Safety

It is of critical importance when creating any exercise related system, to ensure the system is safe for users to use. Shaw et al. identified key health and safety issues that should be considered when developing a virtual reality exercise game. Firstly when designing any exercise game it is critical that the exercise design includes a warmup to prevent injury. Including the warmup as part of the game ensures the users get familiar with the game environment and controls before each intense exercise session. It also means that users are warming up the same muscles that will be used when playing the game (Shaw et al., 2015).

The second factor to consider is that of the cables connected to the head-mounted display. Should the user catch themselves on a cable whilst exercising, there is significant risk of injury. (Shaw et al., 2015). Therefore, we should find a way to ensure that the cables are cleared away from any parts of the user that they could catch on. It is also recommended to keep a one meter clearance around the bike at all times (Shaw et al., 2015).

As we discussed earlier regarding the control scheme (See 2.3.3), sensory disconnects are a major health and safety concern. Some research has indicated that these sensory disconnects or “visual conflicts” may lead to balance related issues such as postural instability and disequilibrium (Redfern, Yardley and Bronstein, 2001). Consequently, we should ensure users are positioned such that there is little risk of them losing their balance and users should always keep their hands on the bike handlebars, something that “PaperDude” neglected to consider (Bolton et al., 2014).

Finally, we have the issue of sweating due to extended periods of exercise. A study found that sweating whilst using the Oculus Rift was not only somewhat unhygienic and uncomfortable, but also sometimes fogged up the screen. This meant that players couldn’t see what the screen inside the HMD (head mounted display) was displaying properly (Shaw et al., 2015). Newer HMD’s should have more breathable insulation and also better venting and seal around the nose to prevent issues such as fogging. However, we should test to see if it is an issue prior to conducting our study.

2.4 The Exercise Protocol

2.4.1 Protocols in Existing Virtual Reality Exercise Games

Of these exercise games we have investigated, it seems like the exercise protocol itself was not pushed as a priority of the project. None of the projects investigated used a well established pre-existing exercise protocol. However, it is worth investigating what exercise protocol they decided upon and the advantages/disadvantages that came with their exercise format.

“PaperDude” did not use a specific exercise protocol, the exercise and game went on simply as long as the user wanted it to. It was simply a tool to assist with whatever exercise the user wanted to do rather than being a regime for a specified duration with any particular intervals (Bolton et al., 2014).

One study decided upon using a short warm up then asking participants to cycle at a moderate-high level for 20 minutes whilst using the system (Shaw et al., 2015). This study had a few issues arise surrounding the use of such a long sustained period of exercise. Firstly was the issue of discomfort as 20 minutes is a significant amount of time to maintain sustained exercise for. Moreover, the added discomfort of having a HMD display attached to the users head for the full duration was not ideal and 1 out of 9 participants had to stop due to nausea. To prevent this, we should aim to use an exercise protocol that takes place over a shorter period of time but with comparable levels of total exertion. This will bring us to using a High Intensity Interval Training Protocol. This should also help with the issue of sweat; since the period of exercise will be shorter, the user will not wear the headset for as long and sweat will have less time to get into the headset (also more modern

HMDs are more breathable).

2.4.2 High Intensity Interval Training

V02 Max is effectively a numerical measurement of an individuals body's ability to consume oxygen. Whilst the body's ability to process oxygen is only half the equation regarding exercise performance (the other half being how efficiently the body uses this oxygen), it is a decent representation of an individuals fitness level.

Extensive literature review has deemed High Intensity Interval Training (HIIT), across many different exercise mediums, to be one of the most effective exercise protocols for improvement as measured by **VO_2 Max**, **T_{vent} (ventilatory threshold)** and **Time Performance** (Laursen and Jenkins, 2002). HIIT was found to improve performance similarly well for both sedentary and physically active individuals. One study found that after 8 weeks of HIIT at 5 sessions per week, users **VO_2 Max** increased by +7%, **T_{vent}** increased by +17% and **Time Performance** increased by 8% (Norris and Petersen, 1998).

A study by Gibala et al. found HIIT to be a particularly effective exercise improvement strategy for aerobic-based exercises. Cycling in particular saw very rapid user improvement in exercise capacity. Their specific study found that after a 2 week period of six High Intensity Interval Training sessions (specifically a Wingate-based protocol), participants exercise time till exhaustion at 80% **VO_2 Max** was effectively doubled (See Figure 2.11 for the results of study) (Gibala and McGee, 2008).

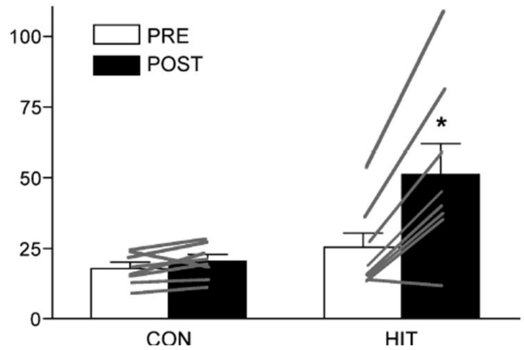


Figure 2.11: Participants time till exhaustion before (PRE) and after (POST) six sessions of HIIT over a two week period vs Participants with No Training Intervention (Gibala and McGee, 2008)

One the most well established and extensively evaluated implementations of HIIT is the Wingate protocol. The Wingate protocol consists of 30 second sprints at near/at maximum capacity, usually separated by roughly 4 minute intervals, for a total of usually 2 - 3 minutes of intense exercise per session (Gibala et al., 2012). Study into the effectiveness of the Wingate protocol vs regular endurance training methods showed a greater increase in participants **VO_2 Max** over a six week period (See Figure 2.12 for results of the study) (Gibala et al., 2012).

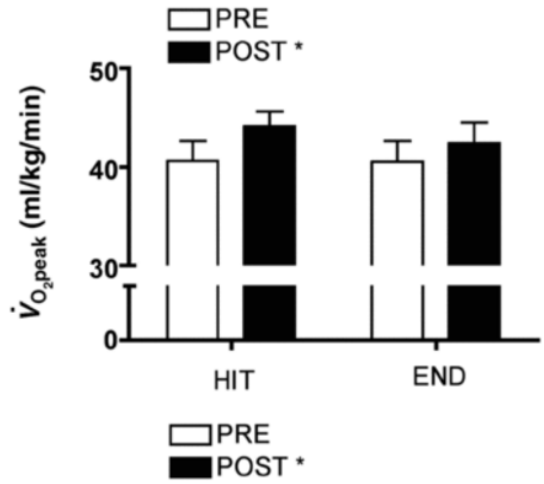


Figure 2.12: Participants **VO_2 Max** before (PRE) and after (POST) six weeks of Wingate-based HIIT over a two week period vs Participants undergoing 6 weeks of Traditional Endurance Training (Gibala et al., 2012)

.

Interestingly, the endurance group had to train for 4.5 hours of consistent exercise per week of the study whereas the directly compared Wingate-based protocol required user to only train for 1.5 hours a week. Moreover, the Wingate-based protocol lead to 10 minutes of intense exercise and the rest was rest (Gibala et al., 2012).

Wingate-based HIIT is, however, extremely demanding and may not be safe, tolerable or appealing for some individuals. As a result, low-volume variations of the Wingate protocol have been devised to be more accessible, safer and more manageable (Gibala et al., 2012). We will use the Wingate protocol as a basis for our virtual reality exercise game exercise protocol, however, we will tune down it's intensity in order to make sure the game is still fun as well as ensuring the exercise protocol doesn't prove to be a barrier to people who usually struggle to maintain/improve at exercise.

Chapter 3

Requirements

Here we determine the requirements for our virtual reality exercise game. We will include requirements for the game design, the hardware/software and a set of brief high level non-functional requirements for the experiment. All requirements have been built from issues and considerations that arose in the literature review. If these requirements are delivered upon, then we should meet the goals of our research project.

The requirements have a fairly large granularity, the overall problem is simply being broken into smaller subproblems that should be solved via the design.

Each requirement will be assigned a **priority** based on the following scale:

- **1**: This **must** be delivered upon, it is an absolute requirement of the specification.
- **2**: This **should** be delivered upon, there may exist valid reasons in particular circumstances to not implement this requirement, but the full implications must be understood and carefully weighed before doing so.
- **3**: This **may** be delivered upon, these requirements are truly optional, different implementations could choose to include or omit these requirements for their own reasons or simply preference.

3.1 Hardware

There requirements are directly related to hardware we are intending to make use of.

3.1.1 The exercise bike must be capable of interfacing fully with the game engine.

Priority: **1** *Functional Requirement*
In order for the in game character to be controlled by the exercise the user is participating in, the bike must be able to pass its current **RPM (Revolutions Per Minute)** to the game engine. Similarly, the bike must be able to receive commands from the game engine to increase its **resistance** in order to provide force feedback as well as define an appropriate level of difficulty for each individual. Resistance will also become a critical part of implementing the "Feedforward Effect". When we refer to resistance of the bike we are quantifying resistance as a breaking force, which in the case of our exercise bike will be the **resistive torque** (Since the forces are rotational) we set the bike to have.

3.1.2 The Head Mounted Display must be capable of tracking users "leaning".

Priority: **1** *Functional Requirement*
The second part of the human control system will be that of leaning. Previous virtual reality exercise game implementations used leaning to control lateral movement of the in game character (See 2.3.3). Previous virtual reality exercise games have made use of the Kinect. However, this hardware had severe limitations (See 2.3.2). Hence, we should instead seek a different modern hardware solution. Newer HMDs such as the HTC Vive and even newer versions of the Oculus Rift both make use of spacial tracking which would allow us to implement a similar system without the need for the Kinect hardware.

3.1.3 The Hardware must be safe to use, hygienic and comfortable.

Priority: **1** *Non-Functional Requirement*
Health and safety is of top priority when we are creating a system for users and carrying out an experiment with human participants. Existing exercise games raised issues regarding comfort and hygiene (See 2.3.3); both of which stemmed directly to the long periods of time participants were exercising for.

3.1.4 We must use an additional separate display so that we can observe the users perspective at all times.

Priority: **1** *Functional Requirement*

It is critical that we are able to see exactly what participant's are seeing within the game; not only do we need to observe their progress and current performance, but we must monitor the health and safety of users throughout (See 2.3.3). If the game experiences any unexpected behaviour we need to be able to react to it immediately, as the users are in a vulnerable state whilst cycling on the exercise bike using the HMD.

3.1.5 We must use some hardware to record Heart Rate.

Priority: **1** *Functional Requirement*

We will want to record each user's heart rate so that we can potentially use it as some measure of user performance. More importantly, we must be able to monitor their current heart rate as it is important in a health study to ensure users heart rate does not exceed their maximum heart rate value.

3.2 Exercise Protocol

These requirements are directly related to the format of the exercise protocol out virtual reality exercise game will incorporate.

3.2.1 The exercise protocol must be accessible and scale to each individuals skill level.

Priority: **1** *Non-Functional Requirement*
We identified lack of accessibility to exercise as a significant contributing factor as to why people struggle to maintain/improve at exercise. Therefore, the difficulty of the exercise should scale to the ability of the individual, if the difficulty is too low users will become bored, and if it is too hight users will be frustrated, both leading to users not being in a state of flow (Csikszentmihalyi, 2014). A lack of flow will result in a similar lack of motivation.

3.2.2 The exercise protocol must deliver an exercise experience based on the Wingate High Intensity Interval Training protocol.

Priority: **1** *Functional Requirement*
We determined the Wingate HIIT protocol to be an effective and very efficient way of helping users improve at physical exercise (See 2.4.2). Phases of high intensity and low intensity could be gamified very distinctively meaning it could lend itself to exercise game format very well. However, it is a very intensive protocol and as it is may not be fully suitable if we want the game to be accessible to a large range of users including sedentary individuals. Realistically we will use it as a basis and alter it based on initial feedback.

3.2.3 The exercise format should be able to be changed to suit future needs/applications.

Priority: **2** *Functional Requirement*
Hard-coding a specific exercise protocol would be simpler, but to extend the life of the game we create and potentially use it in different formats in future experiment, study or real world application. The number of sections and their corresponding durations/intensities should all be able to be altered dynamically.

3.3 Game Design

These requirements are relating to the format of the exercise game as well as the various mechanics that should be used in its design.

3.3.1 The game character must be controlled by the users real world actions.

Priority: **1** *Functional Requirement*

The in game character's forward movement should be controlled by the intensity of the exercise, specifically the RPM of the bike should dictate the speed of the character in game. Similarly, users leaning their head left and right should move their character laterally in a similar way. The control of the character is directly reliant on a reliable interface between the hardware and the game engine.

3.3.2 The game should have suitably challenging short term and long term goals.

Priority: **2** *Non-Functional Requirement*

In order to motivate users via goal-setting, we should try offloading personal goal setting to the virtual reality exercise game (See 2.2.1). If we can provide users with immediate short term goals within sessions (and potentially long term goals between sessions), then we can help keep them motivated whilst using the exercise game.

3.3.3 The game should must engaging game mechanics to ensure users are in a state of flow.

Priority: **1** *Non-Functional Requirement*

In order for users to be in a state of flow and suitably distracted from the exercise they are participating in, there should be suitable short term challenges in place. These challenges should be game mechanics that are giving the user something to focus on at any time to distract them from the simple exercise they are doing. The game mechanics need to be balanced such that they don't detract from the exercise capabilities of the user, but are significant enough to distract them from the exercise they are doing (See 2.3.3).

3.3.4 The game must have an immersive aesthetic with clear distinctions between the games intensity states.

Priority: **1** *Non-Functional Requirement*
Immersion is directly proportional to levels of motivation, as immersion increases users become less aware of the exercise they are currently partaking in (See 2.3.1). The use of particular aesthetics can evoke feelings in users: by making the aesthetic for low intensity sections evoke calm relaxing feelings and the high intensity aesthetic evoke a sense of urgency or pressure, we can directly influence the mindset of users in an effort to improve their overall performance.

3.3.5 The game must include gamification of the motivational factor - "Challenge" (or Self Competition), in the form of a ghost.

Priority: **1** *Functional Requirement*
We identified "Challenge", as the core motivational factor we could directly gamify. Other existing video games and virtual reality exercise games have gamified challenge/self competition in the form of a ghost and seen very positive results. We should use this as our key game mechanic to investigate it's effectiveness, as well as to provide an appropriate medium to make use of and investigate the potential of the "Feedforward Effect", in Virtual Reality Exercise Games.

3.3.6 The game must include gamification of the "Feedforward Effect" to determine it's effectiveness in a virtual reality exercise game.

Priority: **1** *Functional Requirement*
Feedforward and specifically Video-Self Modeling have been proved effective (See 2.6), we intend to investigate the effectiveness of the Feedforward Effect in a virtual reality exercise game. By manipulating the ghost in some way to deliver a concept similar to VSM, we should be able to potentially accelerate users exercise performance improvement.

3.3.7 The player must be able to see the ghost or know how far away it is at all times.

Priority: **1** *Functional Requirement*
Existing virtual reality exercise studies that used ghosts, identified users only being motivated by the ghost when it was in front of them, due to the fact they could not see it when it was behind them and weren't aware of how close or far away it was (Shaw et al., 2016). Our system should ensure that at all times the user can either see the ghost or is aware of how far away from it they are.

3.3.8 The HMD interface must display all relevant exercise statistics to the user.

Priority: **1** *Functional Requirement*

Offloading self-regulatory skills to the game is a good way to help people who lack skills such as tracking of self-progress (See 2.2.1). The HMD interface should display any relevant information that the user may need to keep track of their current exercise state / progress.

3.3.9 The game must be able to record independent variables reflecting user performance.

Priority: **1** *Functional Requirement*

In order to assess the effectiveness of our virtual reality exercise games and the specific game mechanics/concepts we implement, the game needs to record data to help us quantify participants exercise performance.

3.3.10 The game must be able to save and read data between sessions.

Priority: **1** *Functional Requirement*

So that users ghost data and other session specific instances can be retained between sessions, the game must be able to write data (specific to individual users) to files that can be then later be read back by the system. This will be particularly important to ensure consistency in our experiments, as well as being a large part of the underlying ghost implementation.

3.3.11 The game must run at a minimum of 90 frames per second.

Priority: **1** *Non-Functional Requirement*

The refresh rate of modern Virtual Reality Head Mounted Displays (Oculus Rift and HTC Vive) is 90 Hz, thus any fps count above this wouldn't make a visual difference. However if the frame rate falls below 90 fps this can cause significant motion sickness and potentially nausea (Kolasinski, 1995).

3.3.12 The game must avoid sensory disconnects.

Priority: **1** *Non-Functional Requirement*

Sensory disconnects between users real world actions and their corresponding in game actions can cause significant motion sickness and nausea (See 2.3.3). We should aim to map real world actions as closely as we can to in game actions to reduce this effect.

Chapter 4

Design

In this section, we will make use of diagrams to highlight design and, oftentimes, high level and unit to unit interaction.

4.1 Hardware Design

In this section, we will discuss the hardware we intend on using as well as providing an overview of how all the hardware is connected/interfaces with the full system. We will also discuss the specific nuances of each piece of hardware and how this affects our delivery upon the hardware requirements we set ourselves.

See Figure 4.2 for a diagram representing the different pieces of hardware used in our virtual reality exergame and the way in which each interfaces with one another.

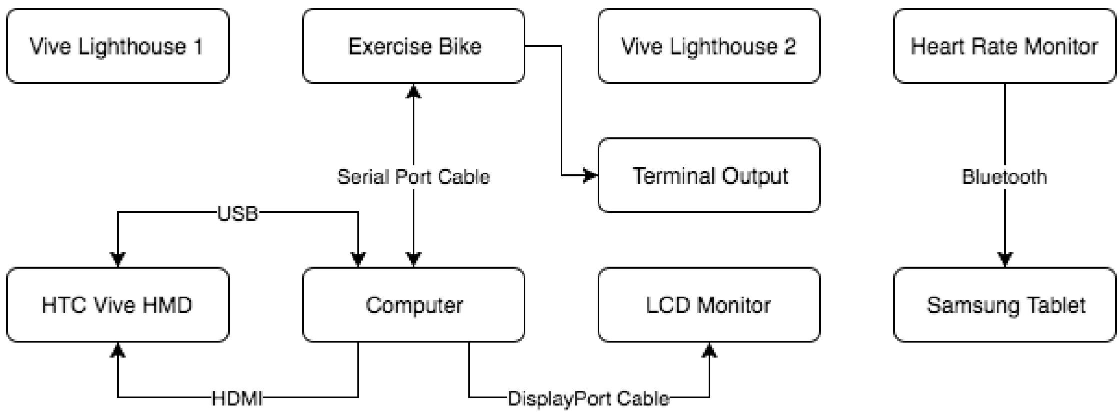


Figure 4.1: An Overview of the Hardware Setup we are using for our Virtual Reality Exercise Game

.

See Figure 4.2 to see the practical hardware set-up in the University DASH Lab.

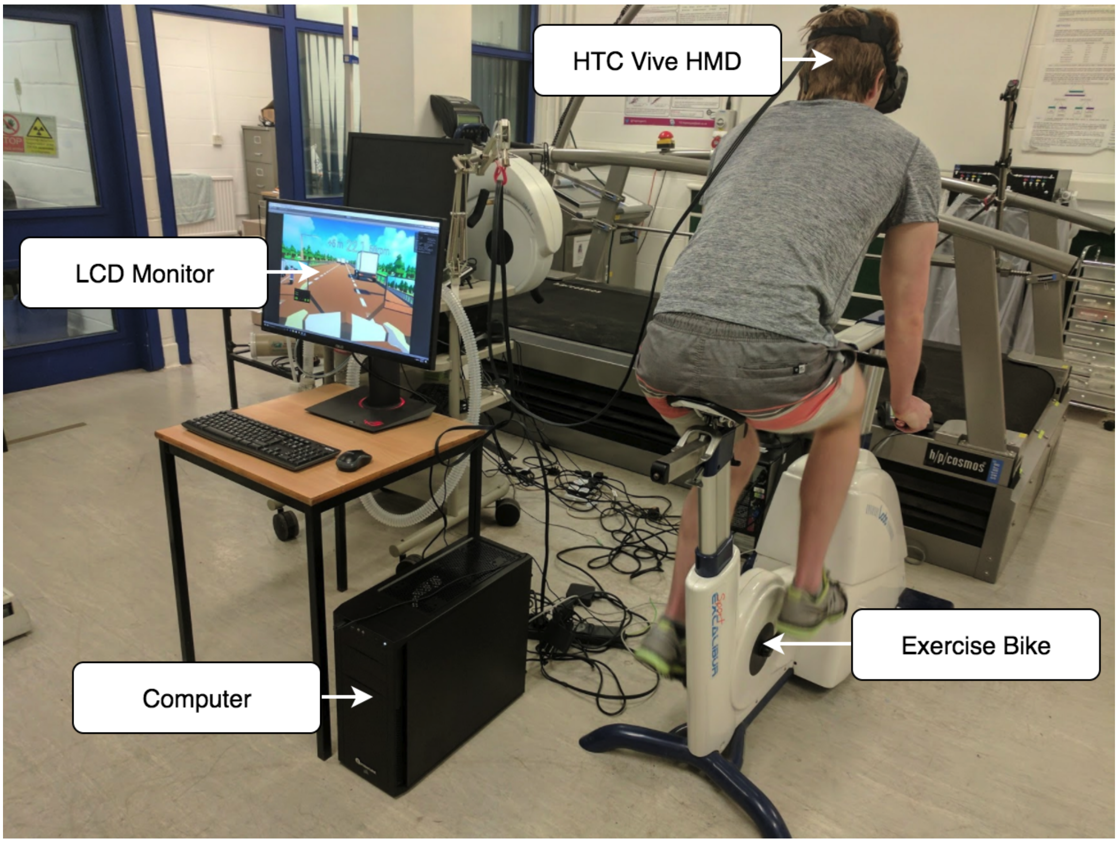


Figure 4.2: The Practical Hardware Setup in the University DASH Lab

.

4.1.1 Computer

In order to run the virtual reality exercise game, we need a computer. To satisfy Requirement 3.3.11, the computer needs to be sufficiently powerful such that it can maintain 90 FPS to avoid motion sickness, whilst graphics quality is at its maximum to provide the most immersive experience possible. To achieve this, the hardware used is very capable and is as follows: an Intel Xeon E5 2680 processor, 64 gigabytes of RAM, and 2 NVIDIA Titan X graphics cards running in SLI (however SLI has no affect on VR performance at the time of writing).

It also has a serial port such that it can interface via serial connection to the exercise bike in order to satisfy Requirement 3.3.8. The system makes use of a wireless keyboard and mouse in an effort to reduce wire clutter that could be a potential health and safety issue if not maintained effectively, which contributes to fulfillment of Requirement 3.1.3.

4.1.2 Vive HMD

The head mounted display we decided to use was the HTC Vive. The HTC Vive interfaces with the PC via two direct cable connections. The visual data is sent to the vive using a standard HDMI 1.4 connection, whilst all other relevant data regarding headset position and orientation that needs to be communicated between the computer and the HMD is sent via USB 2.0 (as we encountered significant issues when trying to interface via USB 3.0 ports). The HTC Vive can track roll, pitch and yaw (See Figure 4.3). We will use roll to control user movement, whilst pitch and yaw will used for the purpose of environment observation. By using roll to to effectively track the users "leaning", the HTC Vive delivers upon Requirement 3.1.2.

To ensure the hardware is safe to use (Requirement 3.1.3), we have used a metal arm with a fabric ring attached to ensure the HTC Vive's cable bundle is always well cleared from the users body (See Figure **??**). To ensure the system both hygienic and comfortable, the HTC Vive uses a soft breathable foam as the seal between the face and the HMD. This foam should prevent sweating to a degree. In the event that the foam needs to be cleaned, it can be easily removed. The soft foam and ergonomic head strap should contribute to use of the HMD being comfortable (Requirement 3.1.3).

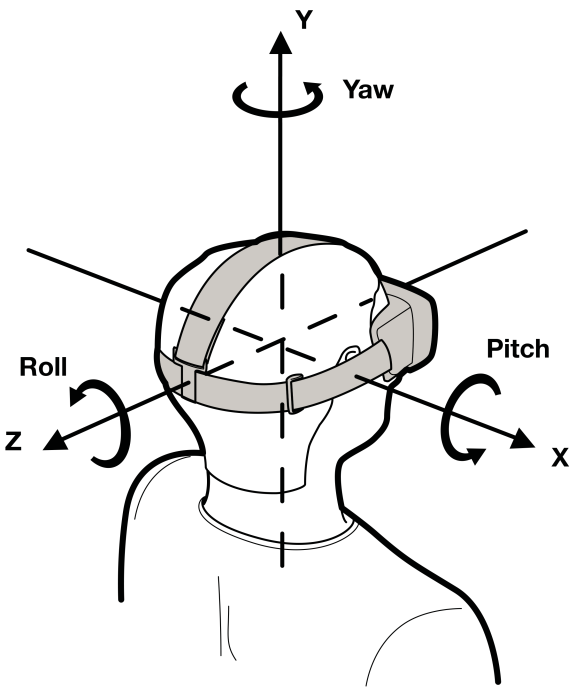


Figure 4.3: HTC Vive Axis Tracking

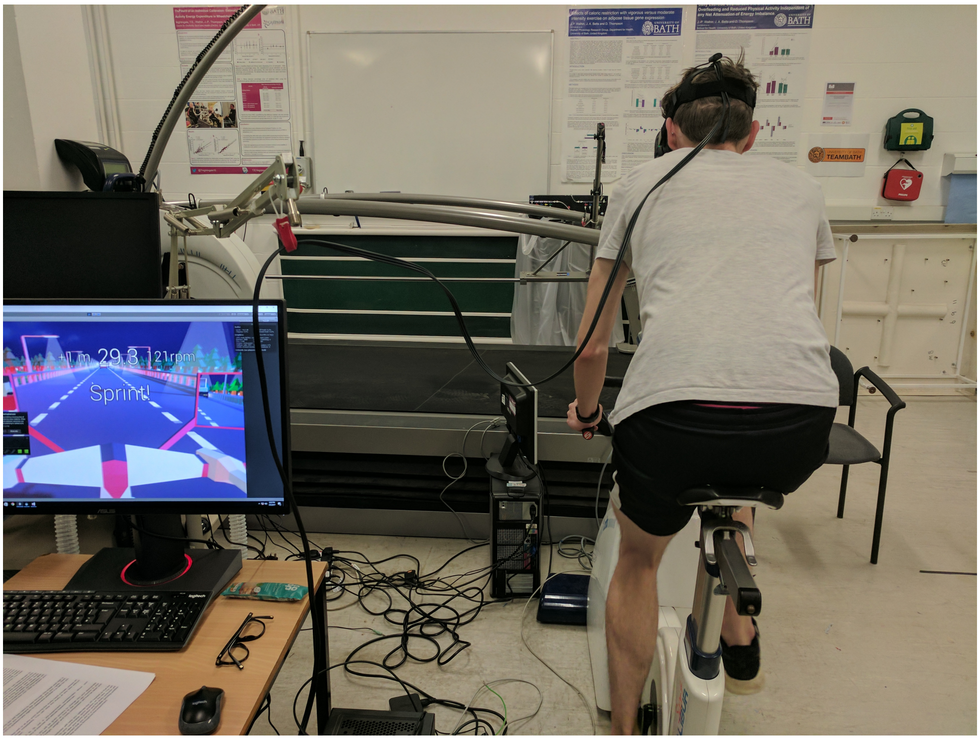


Figure 4.4: HTC Vive Cable Bundle Body Clearance

4.1.3 LCD Monitor

The use of an LCD Monitor in our hardware setup satisfies Requirement 3.1.4. We identified the importance of a separate external monitor for the purpose of seeing what the user is seeing at all times. It also allows us to see what stage of the exercise protocol they are currently at so that the study supervisor is aware of when to ask appropriate questions during the exercise routine. It is also key that the supervisor is able to see if the game experiences any unexpected behaviour, as any unplanned in-game interaction could lead to potential sensory disconnects which we want to avoid (Requirement 3.3.12) for health and safety reasons. The LCD Monitor can be seen in Figure 4.4.

4.1.4 Exercise Bike

The exercise bike is required to be used as the tool for exercise as well as the primary method for in game character control. The bike is a Lode Excalibur Sport; it uses a serial port connection and is directly connected to the computers serial I/O port. The bike is able to directly interface with the computer and in turn directly communicate with our game engine via the 2.0 .NET package, thus fulfilling Requirement 3.3.8. The bike is able to relay its RPM to the computer as well as have resistance changed from the computer both via a specific messaging protocol. We will discuss the specifics in more detail in the Implementation section.

4.1.5 Terminal Output

Directly connected to the Lode exercise bike, the terminal output gives us a direct visualization of the exact values the exercise bike was using at any given moment. These values include the current **RPM** of the bike, the **Resistance** (as a resistive Torque force) and finally the **Power** currently being generated by this RPM and Resistance (See Figure 4.5). This terminal was used for debugging the interface between our exercise bike and the computer and hence critical to the accurate delivery of Requirement 3.3.8.

Figure 4.5: Terminal Output Currently Displaying Resistive Force T and Power P

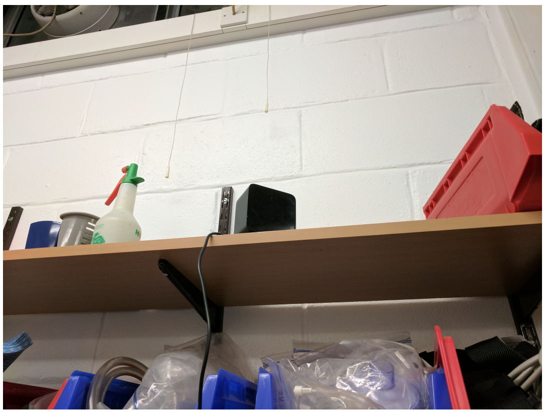


Figure 4.6: The Position of Lighthouse 1

Figure 4.7: The Position of Lighthouse 2

4.1.6 HTC Vive Lighthouses

In order for the Vive HMD tracking to work, the HTC Vive lighthouses need to be plugged in an persistent view of the HMD. Therefore, the working of the HTC Vive lighthouses is crucial to the delivery of Requirement 3.1.2. In order for the best operational success of the HMDs tracking, the lighthouses should be elevated and separated "corner-to-corner" within the square area the HMD is intended to be used in. Since we are using the standing only mode on the Vive, this area need not be large. However, for reliability we have positioned them high and far apart so that people walking through their area does not interfere with the HMDs tracking (See Figure 4.6 and 4.7 to see the positioning of the lighthouses).

4.1.7 Heart Rate Tracker and Samsung Tablet

We are using a Polar H7 heart rate sensor which provides ECG-accurate heart rate readings via low power Bluetooth 4.0. We use this heart rate monitor with a Samsung tablet to make use of the "Polar Beat" fitness application that works in conjunction with Polar H7 heart rate sensor to record graphs of heart rate over periods of time. The use of these two pieces of hardware allows us to fully monitor and record heart rate, thus delivering Requirement 3.1.5. This kind of tracker is far more accurate than more standard optical heart rate wrist trackers such as Fitbits, thus making it a better choice when the hardware is available.

4.2 Exercise Protocol Design

4.2.1 The Standard Wingate Protocol

From our literature survey, we determined that HIIT training would make a good exercise protocol for a virtual reality exercise game based on its scientifically proven effective rates of user improvement coupled with its short and efficient format (See 2.4.2). Existing studies using more standard endurance training protocols are less effective at increasing user performance. When standard endurance training protocols are applied to a virtual reality exercise game format, many problems surrounding the user experience occur. These problems include extended endurance training leading to both discomfort and reduced usability as excessive sweat made the HMD harder to use (Shaw et al., 2015). For these reasons, we are implementing a variation on the HIIT - Wingate protocol to fulfill Requirement 3.2.2.

As previously mentioned, the Wingate protocol can be as unaccessible to users as it is efficient in improving their exercise performance (Gibala et al., 2012). Whilst we want to retain the effectiveness with which the Wingate protocol can improve user performance and deliver upon Requirement 3.2.2, we also want to the game to be accessible to people from both physically active and sedentary lifestyles (part of delivering Requirement 3.2.1). We want to create a system that helps motivate people and helps them improve, but if the exercise regime itself is not accessible to all audiences then it does not matter what game mechanics we create to help encourage people; they will simply fall at the first hurdle.

To tackle this, we will use an alterable High Intensity Resistance to meet the needs of the individual user, contributing to delivering Requirement 3.2.1. We estimated starting values for users based on a scaled down version of "Lode's" (the bike manufacturers) standard Wingate procedure values as seen in Figure 4.8).

Leg ergometry	x body weight = Torque (in Nm)
male adult	0.7 (default setting Lode)
athlete male adult	0.8
female adult	0.67
athlete female adult	0.77
boys (age 7-14)	0.55
girls (age 7-14)	0.53

Figure 4.8: Lode's standard Wingate Protocol Breaking Torque Values for Sections of High Intensity. [2].

We scaled this down to 0.4 times users body weight and then scaled this up/down it to a point where users felt comfortable sprinting such that the resistance was not too overwhelming and too low that their legs span out dangerously (this was important to deliver Requirement 3.1.3). We scaled it down to this level as all the conditions we intended to investigate were being investigated in one experiment session. Therefore, we didn't want users to be exhausted after a single instance of the exercise protocol.

4.2.2 Our Wingate Protocol Variation

We will now give an overview of the different protocols we designed to meet Requirement 3.2.2 and the phases within them, then ultimately which we used for our final study.

This is the flow diagram of our initial exercise protocol design (See Figure 4.9).

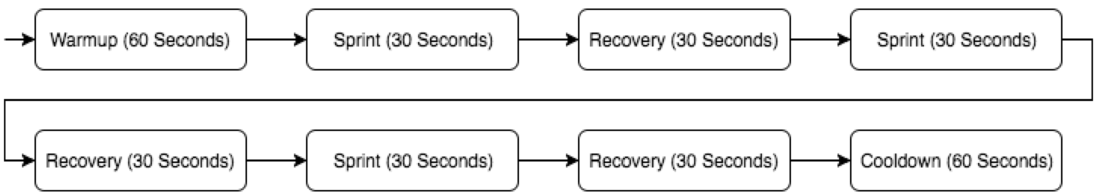


Figure 4.9: Flow Diagram of Our First Exercise Protocol Design

.

[2]http://www.smas.org/2-kongres/wingate.html

We define the different phases as such:

- **Warmup** - The warmup phase is 60 seconds at the beginning of the protocol and uses the low intensity resistance (which is usually significantly lower than the high intensity resistance). We ask users to cycle at around 65-70 RPM as this is a speed that fits well with the game design, doesn't require much effort, but keeps the users legs moving. The purpose of the warmup is to allow users to "warmup" their muscles to prevent Health and Safety issues such as Delayed Muscle Soreness (Olsen, Sjøhaug, Van Beekvelt and Mork, 2012) (this contributes to Requirement 3.1.3). While users could perform the warmup before getting on the bike, studies found that warming up in the game ensures that users are warming up the correct muscle groups (Shaw et al., 2015). Similarly, allowing users to warm up in the game allows them to get acclimated to the game environment/controls before having to commit to a period of high-intensity; if the user is not familiar with the system, then this could lead to dangerous use of the equipment and be a health and safety hazard (Requirement 3.1.3).
- **Sprint** - The sprint is a recurrent phase lasting 30 seconds in which the user should be at levels near "all-out exhaustion". This is the phase we are interested in regarding user performance. In this phase, the resistance is increased dramatically to be the value we discussed earlier at around 0.4 times the users body weight plus or minus an amount to cater to the skill of the user. Within these sessions in the conditions beyond the initial recording, users will be racing the ghost of the first performance. We will discuss this in more detail in Section 4.3.7. The 30 second time frame was similarly based off of Lode's standard Wingate protocol.
- **Recovery** - The recovery is also a recurrent phase lasting 30 seconds. It occurs after every sprint to allow users to recover from the previous phase of "all-out exertion". In the recovery phases, similarly to the other low intensity phases, the resistance is dropped to the low intensity resistance and users are asked to maintain roughly 65-70 rpm. The length of this phase is something we decided to change in our next iteration of our exercise protocol.
- **Cooldown** - Conceptually this phase is once again no different to the warmup or recovery phase, however it is necessary for health and safety reasons similar to the warmup and thus is necessary to deliver upon Requirement 3.1.3. A cooldown period even more so than a warmup allows users to "cooldown" their muscles and has been found to be a key component in reducing Delayed Muscle Soreness (Olsen et al., 2012).

The sprint and the recovery together make up reoccuring **"intervals"**. In our first design, the number of intervals was 3 per exercise session and users were going to execute 3 full exercise sessions during the experiment. After performing a pilot study with a fit male as the participant, we realised that this was simply too much sprinting with not enough recovery time between sprints and the participant became fatigued too early. To tackle this, we went back and redesigned our exercise protocol to accommodate the issue of fatigue when users were expected to carry out the full exercise session 3 times. The iteration of our exercise protocol and our final exercise protocol design can be seen in Figure 4.10.

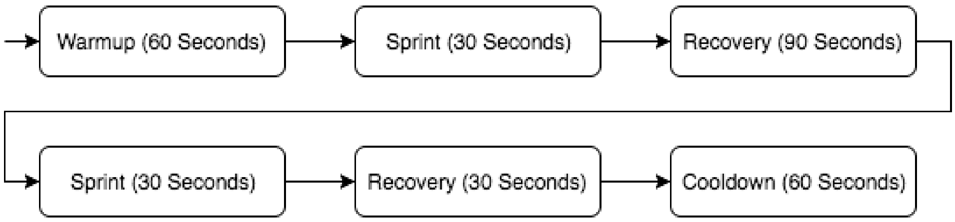


Figure 4.10: Flow Diagram of Our Final exercise Protocol Design

.

After the pilot study, we revisited our exercise protocol design and decided to increase the period of time between the sprints. It's more important for our game that users have a consistent performance as opposed to being fully exerted early with the inability to produce similar results later on. Therefore, we made the duration of the recovery phase between sprints to be 90 seconds long. Since the user was unable to finish all 9 sprints (over the 3 exercise sessions in the experiment) due to fatigue, we decided to try reducing the effect of fatigue by reducing the number of sprints to 2. This still gives us a decent span of time to investigate in our user study whilst reducing the overall effects of fatigue on the later conditions. Aside from these changes, no other changes were made and this was the final design of our exercise protocol for our virtual reality exercise game.

4.3 Game Design

4.3.1 Game Overview

The game contains many design decisions and components so we should start by simply detailing the format and the goals of the exercise game.

- Users control a character in first person in virtual reality.
- They control the forward velocity of the character by cycling and they control their horizontal movement by leaning/tilting their head left and right.
- Users ride along an infinitely long road trying to dodge trucks that are driving straight along the road at a slower speed.
- In the periods of low intensity, users can relax, take in the environment and simply dodge trucks as they cycle along.
- In the periods of high intensity, users are being chased by police and should be cycling at "near all-out" levels while also dodging trucks.

- In the first session, users behaviour is recorded to be used in session 2 and 3 as a ghost.
- In session 2 and 3 users will race against the ghost they recorded in the first session in the high intensity phases with the aim of beating their ghost.
- Once the user finishes the exercise protocol in place, they have completed a session.

4.3.2 Level Design

Users will be racing along a track in this game. We wanted to pick a level format that users are familiar and so we opted for a 3 lane motorway-esque design (See Figure 4.11). It is an environment that users should be familiar with from the real world and also the three lane layout that many existing games have used previously. Some of the most popular 3 lane "infinite-runner" style games in history are mobile games such as Subway Surfers (See Figure 4.12) and Sonic Dash. These two games cumulatively have around 700 million downloads (on the Google Appstore alone), therefore most of the general public has probably been exposed to these games in some form or another so the idea of dodging oncoming obstacles on three tracks should not be hard to grasp or a totally foreign concept. Since we want the game to be accessible to anybody (gamer or non-gamer), drawing game mechanics/formats from the mobile market which caters to a wider more casual gaming audience seems like a good idea.

Figure 4.11: Our Three Lane "Infinite Runner Style" Level Design

Figure 4.12: Screenshots from the hit Mobile Game - "Subway Surfers"

Since the game we are creating will allow variable distances to be traveled in a fixed amount of time, there are a couple of approaches we could use. We could create a very long large map with no potential for variability (so long that users could never feasibly reach the end), or we could create a procedurally generated track that would require chunks to be built ahead of time (therefore, the track is built in real-time as the player rides through the game).

The first method may be easier but it leaves no option for track variance and the form of random track generation between users. Instead, we chose the latter option and decided to have the track built dynamically in real-time. To do this, the first step was to create "**track segments**" using the object assets we created (more on these assets in Section 4.3.6). We would then use these tiles to effectively build a track by instantiating the pseudo-random tile prefab in the location where the next tile should be placed (at the end of the road), as well as deleting the least recently placed tile - i.e. the one furthest behind. This would generate a track that was always a specified number of tiles in length at any period in time as seen in Figure 4.13.

Figure 4.13: A Randomly Generated Track with Fixed Tile Size

.

In our initial design, there were tiles created for left and right turns as well as straight sections (See Figure 4.14 for an example of the tiles we created). However, after testing the implementation of auto turning in our game we decided we would have to leave them out for our experiment due the sensory disconnect making people nauseous which conflicted with Requirement 3.3.12. We will discuss the nature behind the track spawning system and the problems its implementation caused in greater detail in the implementation section.

4.3.3 Short Term Gameplay Mechanic - Avoid the Trucks

We wanted the default task to be enough to distract from the exercise but it should not hinder users' exercise progress or be frustrating. It is key that we get this balance correct to ensure that users are in a state of flow whilst playing our exercise game in order for them to perform at their best and have a positive enjoyable experience playing. Providing a simple yet fun/engaging persistent task to the user should help us reach this state of flow and thusly contribute to the fulfillment of Requirement 3.3.3.

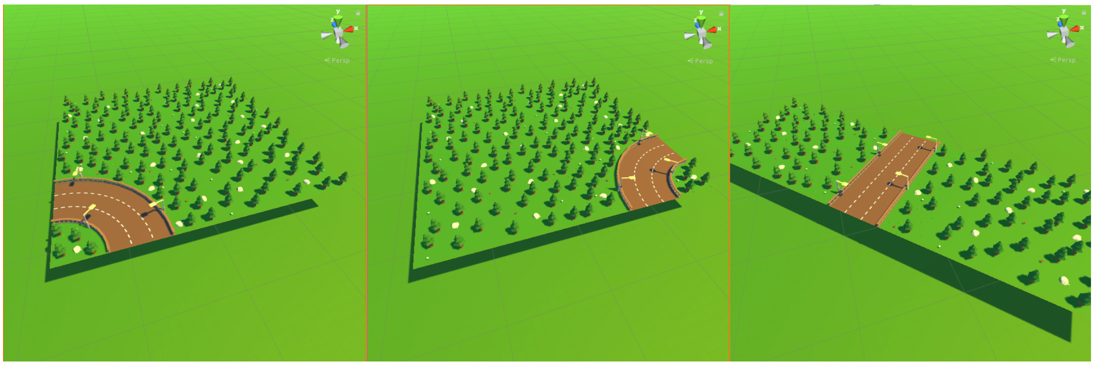


Figure 4.14: Some of the Different Tiles Used by the Procedural Track Generator

.

Other virtual reality cycling games have used the main game mechanic of dodging terrain/objects and reported users having a definitively enjoyable experience (Bolton et al., 2014; Shaw et al., 2016; Shaw, 2014; Shaw et al., 2015). Alternatively, we could have opted for the main mechanic to be collection based. Collection based mechanics are common in many video games (See Figure 4.12 for coin collecting mechanic in Subway Surfers). However, from a virtual reality design standpoint there are some potential issues with using a collection based mechanic.

Firstly, we want the activity to be immersive (Requirement 3.3.4). In some ways, the game should then try to provide an experience that feels intuitive/relatable. In the real world, you would not be driving into objects if you were driving along a road. Hence, implementing this mechanic may be psychologically unintuitive to the user and could potentially affect their performance. Similarly, driving into objects whilst using a Virtual Reality headset could be very disorientating. As player drive into objects, they potentially can fill the screen space and obscure the users sense of awareness in the game world. This issue could be solved by making objects turn transparent or have them potentially being close to the ground as to not obscure the view.

Considering the potential issues regarding a collection based mechanic versus the intuitiveness of an avoidance based mechanic and its proven effectiveness in other studies, we decided to opt for an avoidance based mechanic as the main game mechanic/task that users will have to tackle whilst exercising. Sticking with our theme of driving along a motorway, we wanted to make the objects the user would need to avoid be aesthetically cohesive with the rest of the games theme so we decided to use trucks that users would have to navigate around whilst driving along the road (See Figure 4.16). The avoidance of these trucks provides a suitable, persistent short term goal for the user of "Trying Not to Hit Trucks" to deliver part of Requirement 3.3.2. If users do hit the trucks, they are instantly deleted in game as to not obscure their view and the resistance is subsequently boosted for a short period of time to simulate the feeling of a crash.

Figure 4.15: Trucks the User Has to Avoid Whilst Playing the Game

.

These trucks are not stationary; they are driving down the road at a constant speed. The trucks are instantiated along whenever a new tile tile is created, in a random lane of that tile and then just drive along until they are destroyed by either being too far behind or by being hit by the player/police cars (See Section 4.3.6 for information on the police cars). In order to not hit the trucks, players will have to utilise the control scheme in place to navigate around them.

4.3.4 Controls

The control system has to be intuitive as well as precise. The first half of the control system is the **forward movement** of the player, that is, down the track. As we have already discussed, this is done via the exercise bike. As the user cycles faster, the RPM increases which is interfaced from the exercise bike back to the computer which via a script will store the variable at regular intervals in Unity to be used to determine the forward velocity. This RPM value will be multiplied by some constant factor to translate it into an appropriate forward velocity. This forward velocity will then in turn be used to move the users in game character with which the HTC Vive prefab is attached to in-game, effectively moving both the player and the model forward. This delivers the first half of Requirement 3.3.1 and there is not a suitable alternative solution to this design decision.

The second half of the control system to consider is that of horizontal movement. The user will need a way to move around the track. There are two parts to this design decision that need to be made. Firstly, do we make the in game character turn left and right or simply slide left and right across the track? Secondly, do users control this character movement via the HTC Vives yaw and roll values (See Figure 4.3)?

Shaw et al. created a virtual reality game and tried both methods extensively. The results showed that users found the mapping of leaning to turning left and right unintuitive and caused motion sickness related discomfort (Shaw, 2014). Users much preferred leaning to translate into simple lateral movement, so we decided to opt for a lateral movement solution. Furthermore, if we want to assess performance based on the distance the user travels, then this is not as simple as comparing coordinates since there is now the possibility of users not travelling up the road at the rate they are moving forward, since they don't have to be facing straight up the road.

The only real HMD axis option that makes sense then after choosing to move laterally is that of the **roll** axis. This is because this is the only axis that will allow users to keep facing forward and slide left and right across the road. Shaw et al. found this solution to be effective and intuitive as it closely maps to the way people would move left and right across a straight road in real life by leaning left and right. Previous implementations used a Kinect device, but it was often inaccurate and delayed. The HTC Vive has a near instant response time due to it's Lighthouse implementation, which will greatly reduce sensory disconnects contributing to delivery of Requirement 3.3.12.

We can access the local coordinate space rotation about the HTC Vive's roll axis at any time, even when players turn their head around other axes. Using this angle of rotation, we can create an appropriate horizontal velocity in the correct horizontal direction. This fulfills the second half of Requirement 3.3.1. As users lean their head to turn, we make sure to also lean the in-game character model and bike to give visual feedback that the user is turning.

4.3.5 The Player View and The User Interface

The users in game view is from the first person perspective of the character they are playing as. Users are able to look around the virtual world by rotating around HTC Vive's yaw and pitch axes without it affecting the character control. To ensure delivery of Requirement 3.3.4, the user is able to see the arms and body of their character to help feel immersed in the game. This gives a sense of presence. The head had to be removed from the model to prevent the user seeing it in the HMD. In addition, the full character and bike model tilts when users tilt their head. This simulates the effect of the bike tilting in real life.

In order to make sure the user can see the ghost even when it is behind them (Requirement 3.3.7), as well as see the police cars chasing them, the in-game bike model is fit with two wing mirrors that allow the user to see behind them (See Figure 4.16). The texture on the wing mirrors is a custom shader that effectively generates a reflected image. However, its implementation is not perfect and the reasons as to why will be described in Optimization within Implementation (See section 5.1).

The in game user interface can be seen in Figure 4.17 and contains all the data relevant to

Figure 4.16: The Wing Mirrors Fit to the Bike Model to Allow Users to See Backwards
.

the user we feel needed to be offloaded to the game to meet Requirement 3.3.8.

Figure 4.17: The In-Game User Interface
.

Using the lettered labels from Figure 4.17 here are the details of what each element is and why it was included in the user interface.

- **A** - This is the **Time Left in the Current Phase**. This is a critical bit of information as it allows users to judge how much longer they need to exert themselves for and pick their effort levels accordingly. This value gets larger in size in the last few seconds of countdown to draw users attention to the face the phase is ending. Offloading this key self-regulatory skill should assist users who lack the ability to self-monitor their progress in staying motivated (Dishman, 1991).
- **B** - This is the **Distance Along the Track the User is in Front or Behind their Ghost**. It only appears in the sessions after the ghost has been recorded. This is the final step to ensuring the delivery of Requirement 3.3.7, since the ghost may go behind trucks or fall too far behind so the user needs to be aware of how far ahead or behind it they are even when it isn't in direct sight. It displays a positive number prefixed with a $+$ when the user is ahead and a negative number prefixed with a $-$ when behind.
- **C** - This is the users current **RPM**. This is important so that users can regulate their speed. Without an indication of how fast they are going, users may struggle to pace themselves or go at a desired RPM. This is once again offloading of a self-monitoring skill.
- **D** - This is a line for text prompts, in the code we write we can change this at any point to be whatever text we like to effectively"prompt" users much in the same way a virtual trainer would. The use of virtual trainer systems in virtual reality exercise games has been proven to be an effective method of improving user performance (Shaw et al., 2016). With this we can effectively tell users when they need to start speeding up or slowing down and even when they're nearly done or need to start cycling.

The inclusion of these elements on the user interface fulfills Requirement 3.3.8. The only other piece of data we had access to that we could have included would be the resistance of the bike. However, the resistance isn't used in any way to determine speed so it's inclusion would be wholly unnecessary (We also don't want to include it as we will actually be surreptitiously manipulating it in a later condition to simulate our own version of the Feedforward Effect).

4.3.6 Game Aesthetic

The aesthetic of the game is critical to evoking the correct response and psychological state in our users. Aesthetic is defined as "The way a game looks, sounds, and presents itself to the player." (Niedenthal, 2009).

Many elements have contributed to aesthetic that we have mentioned already. We should use the aesthetic of our game to try to evoke the mental state we want in our users at any given phase of the exercise protocol. Unfortunately, we did not have time to integrate sounds/music into the game and this would only add to the effectiveness of our aesthetic.

As far as this section is concerned, we are simply talking about the way looks and mechanics/situations that add to the atmosphere.

Firstly ,we look at the low intensity phases. In the low intensity phases we want to make the users feel relaxed as their goal is simply to recover or warmup/cooldown. We do not want to add any additional sense of pressure and we want the environment to reflect the cheery relaxed mood we want our users to be in. To do this, we are using a bright sunny scene with high bloom. We tried to focus on using a colour palette full of bright colours that hold connotations of happiness (See Figure 4.18 for the Low Intensity Aesthetic). A nice summer's day is a very stereotypical relaxing scene and this is what we aimed for. To this end, we used a cheerful bright blue skybox.

Figure 4.18: The Game Aesthetic in the Low Intensity Phases

.

On the other hand, we have the high intensity phases. In these high intensity sections we want the user to cycle fast. Firstly, to make sure there is a clear distinction between the high and low intensity sections, we use the complete opposite aesthetic. We change to a night time scene with street lamps turned on and police lights flashing. This aesthetic is not relaxing and should put pressure on the user (See Figure 4.19 for the High Intensity Aesthetic). We want the user to feel pressure to work harder than they would in the calm relaxing sunny scene.

Using the idea of pressure in video games to get people to perform better is not a new idea, even in exercise games. One particular study found using zombies to help push users towards the end of runs to be a great way to get user perform better (Darby, 2014). We attempt to use a similar concept of police cars chasing you in the high intensity sections.

Figure 4.19: The Game Aesthetic in the High Intensity Phases

.

The police cars work in a unique way. We never want the police cars to actually out perform the user (as this would be demotivating in the instances it occurred) but we do want them to get progressively closer over the duration of the high intensity sections. So the police cars always travel at the same speed as the user with a small amount added on that always makes them get to a specifically close distance to the user before the high intensity phase ends (See the difference in police distance between Figure 4.20 and 4.21).

Figure 4.20: The Distance Police Cars are from the Player at the Start of the High Intensity Sprint

Figure 4.21: The Distance Police Cars are from the Player by the End of the High Intensity Sprint

4.3.7 Long Term Gameplay Mechanic - Race The Ghost

We already established the users' short term immediate goals of simply dodging trucks as they drive up the road. However, the main mechanic we want to employ in our virtual reality exercise game and the factor we want to investigate the effectiveness of (with regards to performance and motivation) is that of a ghost.

As we discussed in the literature review, the use of ghosts in video games has been very extensive. There have also been a few virtual reality exercise games that have investigated the effectiveness of users racing their ghost - albeit not in a High Intensity Interval Training format. We also intend to take this idea further by extending it's effectiveness through use of the "Feedforward Effect".

The ghost looks exactly like the player. This is so the player associates themselves with the ghost (See Figure 4.22 to see the ghost in game). It is important that the player is aware the person they are racing is themselves (this is especially important with regards to the Feedforward Effect). The ghost is recorded in the first session and then in subsequent sessions the ghost is played back, effectively showing the user in real-time their **exact** previous performance. The ghost will be present in every phase of the exercise to serve as a constant reminder of it's presence. However, users only have to beat the ghost in the sprint phases of the exercise protocol. To ensure that the race happens on equal footing between the player and the ghost, 5 seconds before the sprint phase the players screen fades to black and they are teleported to the start line and frozen in place (See Figure 4.23). The timer then ticks over into the sprint phase and both the player and the ghost begin moving at the same time to ensure a fair race. We will explain the reasons for this in implementation as this design decision only came about late into development due to issues we had not previously considered.

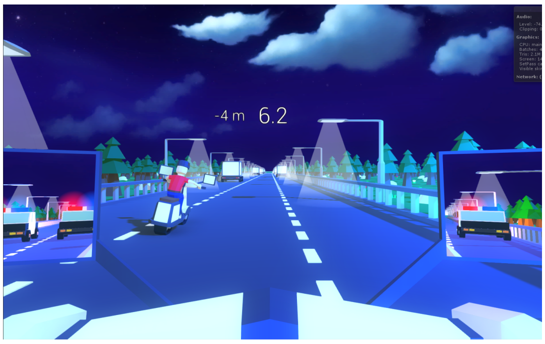


Figure 4.22: The "Ghost" as Seen In Game by the Player.

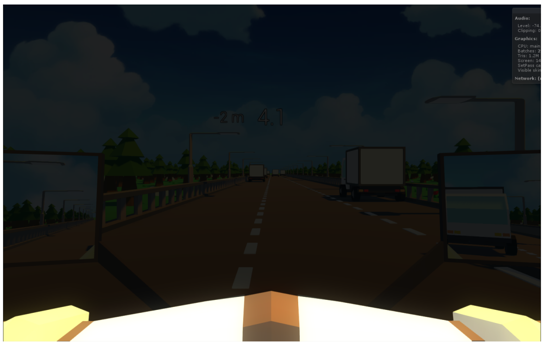


Figure 4.23: Screen Fade to Black Effect, Before Teleporting the Player

By using a ghost, we are hoping that in subsequent sessions we can give users the long-term goal of "Beating Their Ghost". This effectively translates to users improving their exercise performance. The ghost will provide a long term goal to fulfill Requirements 3.3.2 and the gamification of "challenge" as an exercise motivator should act as an interesting mechanic to keep users engaged enough to put them in a state of flow to fulfill Requirement 3.3.3.

There is not much documentation regarding the implementation of ghosts in racing games. However, we came up with two possible options to implement the ghost in our game:

- The first is to **record the players positions** and save these to a file to then be read back and used to position the ghost.
- The alternative is to **record the players inputs** and effectively use the same player controller for the ghost.

Both of these options are viable as an implementation, but we went with the recording of player inputs. The reasons/nuances as to why we determined during implementation thus we will look at them in detail in the Implementation section. An implementation of either ghost design would satisfy Requirement 3.3.5.

We also added functionality to our game that allowed the recording of the players performance to overwrite the old ghost, such that the recorded ghost would always be your best performance. However, for the sake of the experiment we did not make use of this.

4.3.8 Incorporating the Feedforward Effect

The Feedforward Effect and specifically Video-Self Modeling have been shown to help people improve as we discussed in the literature survey. We intend to make use of this concept of showing users performing at a rate higher than they are capable of in real-time through the players ghost. If we can effectively show the ghost performing at a level higher than they have actually achieved, then users may be pushed to perform at a higher level.

The first step is to make sure that players identify with their character. We do this by firstly showing a small introduction sequence at the start of each session in which the camera rotates around the players character and an on screen prompts tells the user "This Is You" as seen in Figure 4.24.

Following the rotation, the camera pans down into the head of the character after which the player's perspective includes the arms and body of the character we just showed them. This helps the user identify as this character and makes the user aware that they will playing as this character throughout the virtual reality exercise game. Once we form a connection between the game character and the user themselves, we can take the next step in making use of the feedforward effect.

The next step is to make the ghost look exactly like the player's character so that the player associates the ghost with their own performance. It may not be enough for the user to simply be aware that the ghost is their previous performance. We want users to believe as much as possible they are always racing against themselves. By making the players character and the ghost look the same, this should reinforce the idea that the ghost performance is truly their performance.

Figure 4.24: A Still Image from the Short Introduction Sequence That Helps The User Identify As The Game Character

.

Up until this point, the Feedforward Effect has not been utilised as we are simply giving them a visualization of their previous performance. In order to make use of the Feedforward Effect, we want to show their previous performance performing at a level **they have not reached yet**. In a similar concept to VSM we hope that if we can alter the user's recordings to show it behaving in exactly the same way but simply performing slightly better, then we can provide users with a real time version of themselves to compete with that is actually operating at a slightly higher level than their true performance. We will investigate whether or not tying the gamification of competition (in the form of a ghost) and the Feedforward Effect together helps users improve at a faster rate.

In terms of the actual design of the feedforward effect there are a lot of nuances that tie into the implementation that dictate the way we choose how to design this effect. Ultimately, there are two real ways we can go about this, either:

- We **speed the ghost up**. In this case, the ghost will retain exactly the same motions and movement but should simply travel at a rate that is some defined percentage faster than the original recorded performance of the user.
- Alternatively, we can keep the ghost the same speed but effectively **make the user work harder**. This design is certainly more risky in terms of being undetected. However, increasing the resistance achieves a similar effect. The user sees the ghost

performing at a level they currently have not achieved as the effort required to maintain the same pace as the ghost has been increased.

Either of these potential designs will fulfill Requirement 3.3.6. We decided to opt for a design that increased the resistance rather than speeding up the ghost. Whilst this condition may be more easily detected due to the feel of the exercise potentially shifting, it incurs far less problems with regards to implementation than a design focused on speeding the ghost up would. We will discuss the exact issues regarding implementation of the first strategy in the implementation section.

4.3.9 The Menu System

The game menus provide a way to navigate between game states as well as provide a way for us to interface with the game, that is, giving us a way to write in variables we want to specify for each session as well as a medium to get results out at the end of a session.

A state diagram illustrating the way the menu system works can be seen in Figure 4.25.

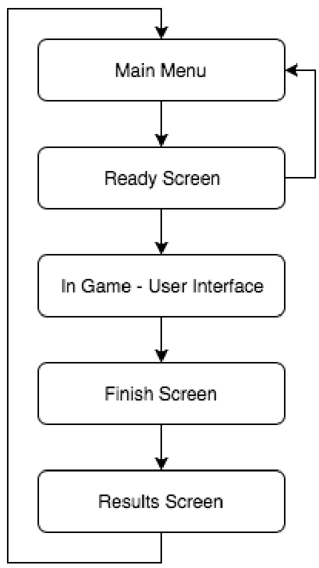


Figure 4.25: The Menu System State Diagram

.

The functionality of each menu is as follows:

Main Menu

The Main Menu can be seen in Figure 4.26. The main menu serves as the starting point for each session.

Figure 4.26: The Main Menu

.

It allows us to input values to be used in the session:

- **UserID** - The unique number each user is given to write the track recording and ghost data to memory and then in turn read it back and utilise it in subsequent sessions. The UserID coupled with some background implementation to read and save data is used to deliver Requirement 3.3.10.
- **Low Intensity Resistance** - The resistance (breaking torque) that is used by the bike in the low intensity phases; for all of our experiments, we used a resistance value of 12 that remained unchanged. However, there is the ability to change the value for every session should we want to.
- **High Intensity Resistance** - The resistance (breaking torque) that is used by the bike in the the high intensity phases. We initially set this to be be 0.4 times body weight and then in the training phase we allow to decide if they want to increase/decrease it based on their preference/physical skill level.
- **Straight Bool** - This value is simply set to 1 to generate a straight track. Any other value lets the game generate a track with curves (assuming there is no existing recording).

The Main Menu leads on to the Ready Screen via the **LOAD GAME** button. Alternatively the game can be exited by pressing the **EXIT GAME** button.

Ready Screen

The Ready Screen simply acts as a buffer between starting the game and inputting the values in the main menu. After moving to the Ready Screen, any data that needs to be loaded into the game regarding the UserID specified is loaded in now. The Ready Screen can be seen in Figure 4.27.

Figure 4.27: The Ready Screen

.

From the Ready Screen, we can navigate back to the main menu if we want to change any values (via the **RETURN** button), or we can start the game (via the **START GAME** button) which begins the game and brings up the user interface which we have already extensively discussed.

Once the exercise session comes to an end, the user interface switches to the Finish Screen.

Finish Screen

This screen simply tells users they are finished and acts similarly to the Ready Screen as a buffer before we open results. This screen can be seen in Figure 4.28.

Figure 4.28: The Finish Screen

.

The main purpose of this screen is to stop the results automatically opening in case the participant has taken off the headset too early so that they don't accidentally see results.

Results Screen

This screen displays the results we will be using to assess user performance (see an example of the Results Screen in Figure 4.29). It shows the ID of the current user (**ID**) as well as the high-intensity resistance that was used (**SR**).

The rest of the data is split into intervals. Before each interval is a timestamp (**ET**) for the time the interval finished in the format HH::MM:SS AM/PM. For each piece of data, what it refers to is determined by the letters and numbers beside the value.

- The first letter is either P or G. **P** means player and **G** means Ghost.
- The second letter refers to what kind of data it is. **D** is the "in-game" distance traveled in that interval and **R** is the total revolutions of the exercise bike done in that interval.
- The last number/letter refers to which interval it was in - **1** is the first interval, **2** is the second interval, and **T** is the total values of all intervals summed together - i.e. the full session.

Figure 4.29: The Results Screen

.

The results screen is built dynamically so if we define a different exercise protocol with more intervals it will work the same way. The results screen deliver Requirement 3.3.9.

4.4 Software Design

In this section, we start by discussing the software we will be making use of and the reasons why we decided upon these. We will follow up by giving the top level overview of the games script structure

4.4.1 Software Choices

Game Engine

The first decision to make was that of Game Engine. Ultimately, any good game developer would be able to create a small scale project like this in any of the popular Game Engines such as Unreal Engine 4, Unity3D, or even by building our own Proprietary Engine.

The short time constraint was a major factor for our consideration. This immediately ruled out creating our own Proprietary Engine as this was a lot of extra effort and would end up becoming a large focus of the project. This leaves a choice between the more standard widely utilised game engines - specifically Unity3D and the Unreal Engine. Regardless, the project could be delivered to a good caliber using either of the two game engines.

Unity3D is known for its rapid deployability and ability to create projects for all platforms with ease. Unity utilises JavaScript or C# as its scripting languages. Unreal Engine on the other hand is arguably better at achieving more realistic graphics/particle simulations and has access to a blueprint system for users who do not want to program in C++.

In the end, we simply decided on the platform we were most comfortable with. We had significant experience using the Unity Engine and felt comfortable that we could achieve the cartoon-esque aesthetic we desired using this game engine (The Unreal Engine has a habit of making everything look quite "realistic"). We felt confident using either C# or C++ and hence language choice was not a deciding factor.

Programming Language

After deciding on Unity3D as our Game Engine of choice, we were left with the decision of using JavaScript or C#. While both can achieve the same end product, we had significant experience using C# in Unity3D and so once again made this decision based on previous experience with the intent of saving time in the development process.

Modeling Software

In order to achieve exactly the aesthetic we were looking for, we manually created all of our own assets (excluding the sky-boxes). Every model in the game was created using 3D modeling software. We chose Blender as our modeling software chiefly due to previous experience and the short time constraint upon us.

4.4.2 Script Structure

Here we look at the script structure of the virtual exergame, then discuss briefly the purpose of each script and what it needs to interface with.

Firstly, the overall structure and interfacing between scripts can be seen in Figure 4.30.

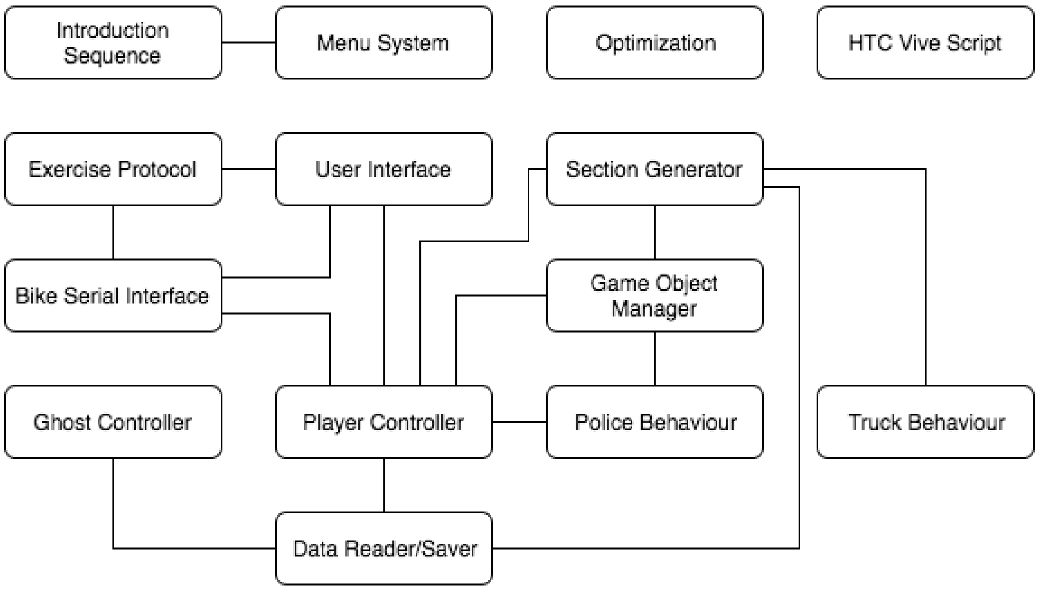


Figure 4.30: The Script Structure Diagram

.

Here is a brief overview of what each script does and why it interfaces with the scripts it does:

- **Introduction Sequence** - This provides all the functionality required to animate the camera to create the initial introduction sequence at the start of the game.
- **Menu System** - This provides all the functionality relating to Unity Screens to create our full menu implementation. It starts the introduction sequence so needs to interface with it.
- **Optimization** - This contains any of the additional functionality required to optimize our game to ensure it runs at 90 fps, thus delivering Requirement 3.3.11.
- **HTC Vive Script** - This script is an asset downloaded from the asset store simply to support full functionality of the HTC Vive in Unity.
- **Exercise Protocol** - This is used to define the exercise protocol as well as track times and switch intensity states. This script provides functionality to deliver Requirement

3.2.3. It needs to write the resistance to the exercise bike when changing intensity states hence this interface.

- **User Interface** - Displays all the relevant data to the user on the HMD interface. It will need to pull RPM from the Bike Serial Input Script, the time from the Exercise Protocol Script and ghost separation from the Player Controller hence these interfaces.

- **Section Generator** - This script will generate a new segment of track as well as delete the oldest segment, it does so similarly with trucks. It needs to read the recorded track/truck positions from previous sessions from the Data Reader/Saver and also needs to calculate some values for trucks after instantiating them using Truck Behaviour, which is why it interfaces with these scripts.

- **Bike Serial Interface** - Contains the code to interface with the exercise bike via a specific protocol. It is used by other scripts to interact with the bike.

- **Game Object Manager** - Builds and stores list of instantiated game objects. It is used in the managing/deleting of game objects on a mostly first-in-first-out basis, when procedurally generating the level. It is interfaced with other scripts that need to delete objects from the global lists.

- **Ghost Controller** - Defines the ghosts behaviour based on the recorded player inputs. It pulls player inputs from the Data Reader/Saver to move the ghost.

- **Player Controller** - Uses the Bike Serial Interface and the HTC Vive prefab object to determine the players movement, it also defines the players interaction with other scene objects. It interfaces with the Bike Serial Interface to achieve this. It interfaces with the Section Generator to spawn new sections when the player enters each tile, also it interfaces with the Game Object Manager to delete trucks the player crashes into. It sets the police cars speed, by interfacing with Police Behaviour and finally interfaces with the Data Reader/Saver to save the players inputs for the session to be used by the ghost in alter sessions.

- **Police Behaviour** - Defines the behaviour of the police cars. It needs to interface with the Game Object Manger to delete trucks that the police cars pass through.

- **Truck Behaviour** - Defines the behaviour of the trucks.

- **Data Reader/Saver** - Handles the reading in of data and saving of data specific to User ID's. This script provides functionality to fulfill Requirement 3.3.10.

Chapter 5

Implementation

In this section we look into the most interesting areas of our implementation, as well as the problems and decisions surrounding these areas of implementation.

5.1 Optimization

Optimization is one of the most important implementation factors to consider. If the game does not run at 90 FPS (Requirement 3.3.11), then it will certainly induce motion sickness in users (Kolasinski, 1995). When considering optimization, we must look at multiple areas and components of the game that have the biggest impact on performance.

5.1.1 Mesh Combining

The first area for consideration is that of keeping draw calls to a minimum. Unity refers to draw calls in its most recent versions as “batches”. In general terms, for each object in the scene that is visible at any time, Unity will need to perform an additional draw call per texture the mesh has. Our scene will often have thousands of objects in the form of the dense foliage and trees on each tile. Therefore, if we do not employ some technique to reduce draw calls/batches, then the game will be severely slowed down from drawing thousands of times for each frame.

Unity has built in strategies to reduce batches via “static and dynamic batching”. By marking objects as static in the scene, Unity can automatically statically batch meshes with the same material into a single mesh, allowing hundreds of objects to be combined at **build-time** and reducing draw calls significantly. These strategies would be a simple and effective solution but have the limitation of only occurring at build-time. Unfortunately, our environment is procedurally generated at run-time. Hence, objects marked as static are spawned in post build-time and would never be statically batched together.

Unity’s second solution is dynamic batching. This method is not as effective as static batching, but can be used to batch non-static objects in the scene. Dynamic batching works similarly to static batching, however there is a hard limit on the size of the combined meshes it can create. This limit is 900 vertices which is not very large. Dynamic batching would certainly improve our games performance, but by considering the sheer amount of objects that will be in our scene at any moment in time, it is clear that this improvement would not be substantial enough. Dynamic batching in our non-optimized scene managed to reduce the number of batches by over 80% (as seen in Figure 5.1), however the value is still huge and will greatly effect performance.

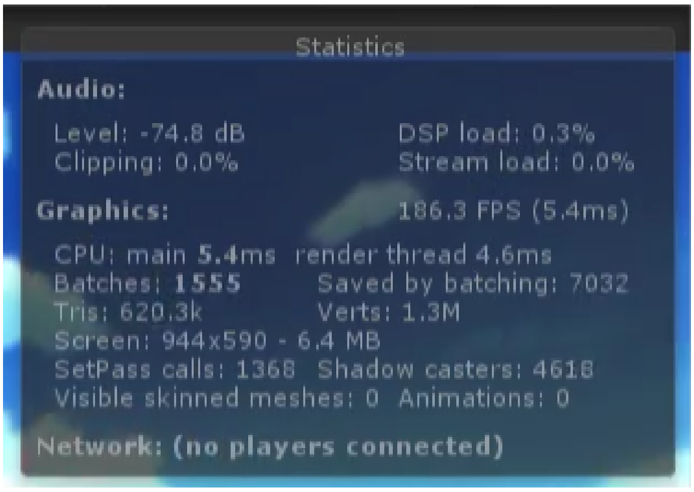


Figure 5.1: Non-Optimized Environment Batch Count

.

Our best option is to try implement our own static batching solution but at run-time. When a new tile is instantiated, our implementation attempts to take all of the children object meshes with the same texture within that tile and combine them into a single large mesh. However, there are some nuances to the implementation and some issues we ran into.

The first step is to effectively get the mesh renderer component from every child within the object this script is applied to and then iterate through every child and extract out all the unique materials (identified by their string names). It does this by using a dictionary to store keys pertaining to specific materials then checking if the material already exists in the dictionary. The code can be seen in Listing 5.1. For each material it creates a list of combine instances and at the end of the next stage, this list will contain combine instances (specific meshes used for combining meshes) of every child object.

Listing 5.1: Mesh Combiner - Material Dictionary Creation

```
Matrix4x4 myTransform = transform.worldToLocalMatrix;
combines = new Dictionary<string, List<CombineInstance>>();
namedMaterials = new Dictionary<string, Material>();

//Get all mesherenderers in children objects
MeshRenderer[] meshRenderers =
   GetComponentsInChildren<MeshRenderer>();

foreach (var meshRenderer in meshRenderers)
{
  foreach (var material in meshRenderer.sharedMaterials)
    if (material != null &&
       !combines.ContainsKey(material.name))
    {
    //Adds the unique meshrenderer materials to the lists
      combines.Add(material.name, new
         List<CombineInstance>());
      namedMaterials.Add(material.name, material);
    }
}
```

We then need to find all the mesh filters in the children objects and we similarly iterate through. This time, for each mesh filter we create a new combine instance (a specific kind of mesh only for combining meshes together) of the existing objects mesh. We check three cases; the mesh filter not having a corresponding mesh, the mesh uses multiple materials, or the mesh doesn't have a material at all. In these cases, the mesh cannot be combined (so they are skipped via continue).

Once the combine instance has been created and given the global transform of the object it was derived from, it is stored in the list that is paired with the mesh's corresponding material in the combine dictionary. This means we end up with a dictionary containing every material and for each material a list of all the meshes that use only that material so that we can combine them together. Once the combine instance is created and stored, we can destroy the original meshfilters renderer as we do not want to keep rendering the old uncombined mesh. This can all be seen in Listing 5.2.

Listing 5.2: Mesh Combiner - Combine Instance Grouping by Material

```
MeshFilter[] meshFilters =
   GetComponentsInChildren<MeshFilter>();
foreach (var filter in meshFilters)
{
    if (filter.sharedMesh == null)
        continue;
    var filterRenderer = filter.GetComponent<Renderer>();
    if (filterRenderer.sharedMaterial == null)
        continue;
    if (filterRenderer.sharedMaterials.Length > 1)
        continue;

    //Creating a new combine isntance (a kind of mesh used to
       combine meshes)
    CombineInstance ci = new CombineInstance
    {
        mesh = filter.sharedMesh,
        transform = myTransform *
           filter.transform.localToWorldMatrix
    };
    combines[filterRenderer.sharedMaterial.name].Add(ci);

    //Destroying the original mesh renderer
    Destroy(filterRenderer);
}
```

The last step in this process is to now iterate through each material in the named material dictionary. For each named material key, we combine all of combine instance meshes in its corresponding list (this list contains all the child meshes that use that material).

To do this, we create a new game object (becomes the new combined mesh) and assign its parent to be the object with the combine children script attached. We then set its local transforms (position scale rotation etc.) to origin/default, so that all the combine instances are in the right location. The new game object is given a mesh filter component which is set to be the combined mesh of all the mesh filters in the list. Then, the game object is given a mesh renderer (to render the material) and the mesh renderer material is to set to be the corresponding material. Therefore, we have combined our meshes and we could have reduced thousands of potential draw calls to just one (since all of our objects use the same material).

Listing 5.3: Mesh Combiner - Creating the Final Combined Mesh

```
//Creating the new combined mesh from all the children meshes
   and rendering just that mesh.
foreach (Material m in namedMaterials.Values)
{
    var go = new GameObject("Combined mesh");
    go.transform.parent = transform;
    go.transform.localPosition = Vector3.zero;
    go.transform.localRotation = Quaternion.identity;
    go.transform.localScale = Vector3.one;

    var filter = go.AddComponent<MeshFilter>();
    filter.mesh.CombineMeshes(combines[m.name].ToArray(),
       true, true);

    var arenderer = go.AddComponent<MeshRenderer>();
    arenderer.material = m;
}
```

This implementation was almost complete but a significant issue presented itself. Originally, we had the meshes that are used for the light beams from the lampposts enabled when the tile was spawned at night time. This lead to the mesh combiner combining the light meshes into a combined mesh which meant the original mesh filters that the light beams used were destroyed. This meant that we no longer had a reference to the light beam meshes that we needed to turn them off when the intensity switched from high to low. This meant the light beam meshes were stuck on in the day time if they were instantiated at night time.

To combat this, we disabled the light beam meshes in the prefab so that when they were instantiated the mesh was disabled and thus not combined into a single mesh. We then made the mesh combiner turn the light beam meshes back on after combining the rest of the meshes if it was night time. This can be seen in Listing 5.4.

Listing 5.4: Mesh Combiner - Creating the Final Combined Mesh

```
void turnOnLightsAfterMeshCombine()
{
    //Iterating through and turning on lights if it is night
       time.
    //We do this because we don't want to combine the light
       beam meshes because we wont be able to turn them off.
    GameObject controller = GameObject.Find("IESController");
    if (!controller.GetComponent<IntensityController>().day)
    {
        GameObject lamps =
           transform.Find("Main/Lamps").gameObject;
        foreach (Transform light in lamps.transform)
```

```
        {
            light.transform.GetChild(0).gameObject.SetActive(true);
        }
    }
}
```

After implementing this run-time mesh combiner, we saw batches reduced to amounts 10 times lower than that of static batching alone (see Figure 5.2). This brought our batching per frame to below 200 which was much more manageable and is the main way we delivered on Requirement 3.3.11.

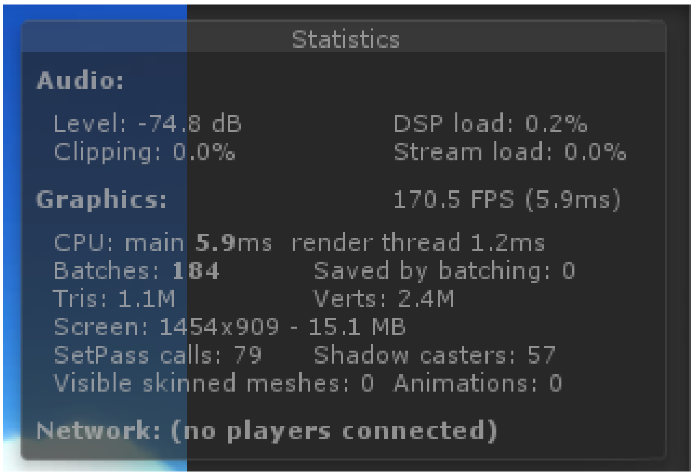


Figure 5.2: Optimized Environment Batch Count

.

5.1.2 Mirrors Implementation

The mirrors were another part of the implementation that we had to consider with regards to optimization. Generally, raising the quality and usability of mirrors in video games dramatically reduces performance. This is why many games do not implement mirrors properly.

When mirrors are implemented, effectively one has to render an entire scene again the mirror. Some methods are far more accurate than others. In our case, we simply used implementations from public resources online. Both methods we implemented unfortunately had their problems. As a result, we were not able (within this time frame) reach a perfect solution.

Both methods worked by creating a custom shader and material and drawing a texture to that material via a script, thus rendering the material to a plane on the wing mirror. Any problems with efficiency that we were going to have would be doubled since we were using two separate wing mirrors which both needed to generate their own texture.

The first method was an adapted version of one of the Unity Standard Water Shaders, called Mirror Reflection Shader 4. We will not focus on the technical details as most of the implementation was already done, however we will discuss the benefits and disadvantages of this implementation. This method was very efficient; it was very easy to scale down the quality of the texture and, even at higher resolutions, the impact on performance was not detrimental which was a significant positive.

However, the reflections were generated on a per-eye camera basis meaning that the texture was different for each eye. This kind of implementation is non-stereo rendering and, as a consequence, the two images (one for each eye) will not necessarily form a single image when viewing both eyes at the same time. This means that when using the Vive HMD, each wing mirror looked like it had two different images overlapping one another. This makes it somewhat hard to focus on the mirrors when using the HMD.

Considering the disadvantages of this implementation, we sought out a stereo rendering solution. This implementation is actually provided as an asset made officially by HTC (the makers of the specific HMD we were using). With regards to the reflection this method was perfect. However, the performance was hit dramatically. Even with drastic reduction in quality and turning off pixel lighting when calculating the reflected texture, it was never quite enough to bring the frames per second above 90. This was a huge issue as it very quickly started causing motion sickness.

Therefore we had to go with our first implementation even though the reflected image did not look quite right. If we went with the second implementation, we would not satisfy 3.3.11 and hence make the game unusable. It is still early into the Vive's timeline and hopefully in the future works for this implementation will give rise to something more efficient. Unfortunately, we did not have the time nor shader expertise to edit the shader to a degree of usability for this project.

5.2 Interfacing with The Exercise Bike and Longitudinal Player Control

Making design decisions for the bike was relatively easy. However, implementation was not such a simple process. There were an assortment of nuances to interfacing with the bike and we will go through the exact process here.

The first step was to determine the exact way we could interface with the bike. We knew the Lode Excalibur Sport used a serial port to read/write to and from. However, we had no indication of the messaging protocol it used to change and read these values. We made use of an existing piece of software that the University Health Department used to specify values for **power** to the bike and to read out **RPM** in their studies. We reverse engineered the protocol by using a serial analyzer.

The serial analyzer was used to analyze the data flow over the serial port connection. By looking at the data flow, we were able to determine the specifics of connecting with the exercise bike as well as the messaging protocol used to read **RPM** and write **Power**. The initial connection establishment was completed via Listing 5.5. Firstly, we create a byte array of size 5 that we use as a read buffer to store the incoming messages we receive. Then, we create a new serial port connection with parameters as seen in Listing 5.5 and subsequently open the port.

Listing 5.5: Exercise Bike Serial Interface - Setup

```
byte[] readBuffer = new byte[5];
bikePort = new SerialPort("COM1", 9600, Parity.Even, 7,
   StopBits.Two);

Debug.Log("Port O");
bikePort.Open();
Debug.Log(" O Port");
```

The messaging protocol uses a simple write/read system, that is, if you ever want to read out a value, you must first write to specify the value you want to receive and then read the value afterwards. The reading of the RPM from the bike is handled by the code in Listing 5.6. The message to write to receive RPM from the bike is **"0,RM.\r\n"**. After writing this we read the next 5 bytes with an offset of 0 and write it into the read buffer. We subsequently decode the bytes into a string as they are parsed in UTF8 format. The first three letters are simply "RM" the rest of the string is the integer number we want as a string. Therefore, the final step to getting the RPM value is to convert the substring from index 2 onwards to an integer. We use a try-parse to handle as the value is not correct in some edge cases.

Listing 5.6: Exercise Bike Serial Interface - Reading RPM

```
bikePort.Write("0,RM.\r\n");
bikePort.Read(readBuffer, 0, 5);

String a =Encoding.UTF8.GetString(readBuffer);
string stringRPM = a.Substring(2);
if(int.TryParse(stringRPM, out rpm))
{

}
else
{
    rpm = 10;
}
```

Based off the existing software we had access to, we could find no way to set resistance (breaking torque) despite the fact we could see the value on the terminal output. We could, however, set the **power** value. Power is the energy being generated by cycling against the breaking torque in joules per second. That is, if we set power to be constant, the faster we cycle the lower the resistance is set to be as the resistive force needed to generate the same power is lower since we are doing more rotations per minute.

We could not set resistance, yet this was the value we wanted to keep consistent within our sessions and one of the independent variables we were intending on manipulating. Therefore, we had to find an alternative way to set the resistance. Since we could set the power level and we knew that breaking torque was automatically set by the bike to equal the amount needed to generate the specific power level at any RPM, we could effectively trick the bike into consistently setting the resistance to a value we specify by sending regular power writes based on the current RPM.

We determined the equation to work out what this power level should be to be:

$$Power(Js^{-1}) = Resistance(Nm) \times RPM \times constant$$

We used trial and error to find the value of the constant that we needed to use for the resistance to be correct on the bikes terminal output. This value is 0.11.

The final piece of code that implements this can be seen in Figure 5.7. The only extra piece of information here is that of crash resistance - this is a value thats gets set to an amount above one when the user crashes into a truck and then quickly falls off back to 1 (the default). This simulates a burst of resistance to provide force feedback for the crash.

Listing 5.7: Exercise Bike Serial Interface - Writing Resistance Via Power

```
//Calculating the power level to set the bike to based on the
   desired resistance and the current rpm
float powerFloat = (resistance * crashResistance) * rpm *
   constantVal;
//We floor the value since the bike only accepts integer
   values.
totalPower = Mathf.FloorToInt(powerFloat);
bikePort.Write("0,SP," + totalPower + "\r\n");
```

In order to use the values, other scripts are able to read the RPM value that the script stores every time this loop executes as well as set the resistance value that this loop uses.

The next element of the bike interface to consider is how often to execute this loop. The loop is in a `while(true)` that is within a coroutine which yields the **waitforseconds** method that takes a float amount of seconds to wait before effectively executing the loop again. We found that we were unable to execute this loop in every **fixedupdate** (which is the unity global shared thread that gets called exactly 50 times per second); it crashed the game when we tried to run it this often. We found that we could execute this serial loop 10 times a second without it impacting the game performance greatly. This value is sufficiently small to make the game feel incredibly responsive (which is important to ensure there is no sensory disconnect and we meet Requirement 3.3.12).

The RPM value that this code determines is used very simply by the player controller. Firstly, RPM is divided by a constant to become a **speed** value which we store for use by the ghost and then the controller determines velocity using this equation:

$$LongitudinalVelocity = Speed \times constant \times Transform.forward$$

The value of the constant does not matter as it has no significance without considering the scale of the game. Therefore, we just chose a value that gave us the sense of speed we wanted in game.

5.3 Interfacing with HMD and Lateral Player Control

In comparison with implementing the bike interface and longitudinal control system, interfacing with the HMD is very easy. We simply import the asset pack made to support the HTC Vive, then placed their prefab for the HMD setup in our scene.

With this prefab in the scene, we now had a game object with the appropriate cameras attached for use by the HMD. By accessing the z-component of the game object's transform component Euler angles, we can effectively get the rotation around the HMD's real-world **roll** axis (this is the user tilting their head side to side as seen in Figure 5.3).

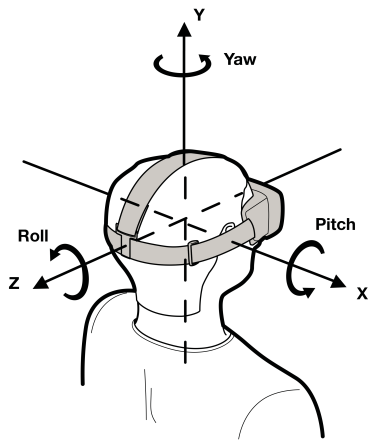


Figure 5.3: HTC Vive Axis Tracking

This is because Euler angles of an object's transform represent its rotation relative to the global coordinate space. The Euler angles represent the rotation around x (pitch), y (yaw) and z (roll). Any possible object orientation can be represented by some amount of rotation about these axis separately.

As can be seen from the image, tilting the head to the left will yield a positive rotation whilst tilting the head to the right should yield a negative rotation. However, Euler angles are never negative. Euler angles lie in the range [0,360]. For example, the user tilting their head -10 degrees about the z-axis would give a Euler angles z component of 350 degrees.

In our implementation, we want to use the positive and negative values rather than just positive rotation. If the user is leaning to the left, their Euler angle will be greater than 180 with the current system. Considering this, we use the following code (Listing 5.8) to get all angles as the closest positive or negative rotation about the z-axis.

Listing 5.8: Head Mounted Display Rotation to Calculate Latitudinal Velocity

```
float rotation = headset.transform.eulerAngles.z;
//This is the case when we have horizontoal rotation rather
   than being between 0 and -90 the value is between 359.999
   and 270
//So we take away 360 when it is rotated this way to get the
   negative value we want.
if(rotation > 180)
{
    rotation -= 360;
}
lean = -rotation / maxLean;
```

We divide the rotation value by a constant maxLean to get a value that we store to be used be the ghost (**lean**). We then use the lean value in a similar way that we used RPM to calculate the players latitudinal velocity:

$$LatitudinalVelocity = lean \times constant \times Transform.right$$

The constant value was selected so that the movement felt responsive and close to real-life simply by trial and error.

In both determining longitudinal velocity and latitudinal velocity, there was no need to use two different constants to achieve the end result. However, we did so for the sake of debugging.

5.3.1 Data Saving/Loading

Data saving and loading is crucial to provide the functionality of the ghost by saving and then later loading recordings of user inputs. It will also be used to save the positions that trucks were spawned in and the exact track tiles spawned to provide a consistent experience between sessions when testing.

There is a single public static instance of the DataSaver class instantiated within the class itself (called saver) which means that this instance of DataSaver can be accessed from any script with ease - simply by using **Datasaver.saver**. On top of this, we make sure to give the object the DataSaver script is attached to the DontDestroyOnLoad property which means that even between loading different scenes the DataSaver instance will persist. This can all be seen in the code in Listing 5.9.

Listing 5.9: The Datasaver Singleton

```
public class DataSaver : MonoBehaviour {

    public static DataSaver saver;

    //Unsaved variables
    public string userID;
    public int lowResistance;
    public int sprintResistance;
    public bool straight;
    //Saved variables
    public int[] trackRecording;
    public int[] truckRecording;
    public float[] ghostSpeedRecording;
    public float[] ghostLeanRecording;

    void Awake () {
        DontDestroyOnLoad(gameObject);
        saver = this;
        }
```

The Datasaver class has a method for saving and a method for loading. In order to save data, we need to create a serializable class to serialize to a file. When reading, we can deserialize the file to get out the class we put in with all the variables it contains. The serializable data class that we we will write to files can be seen in Listing 5.10.

Listing 5.10: The Serializable Class to be Written To Files

```
//A serializable class we use to store all the data we want
   to save/load in.
[Serializable]
public class userData{

    public int[] trackRecording;
    public int[] truckRecording;
    public float[] ghostSpeedRecording;
    public float[] ghostLeanRecording;
}
```

We save/read 4 different pieces of data using this data structure:

- **Track Recording** - This is an array of recorded integer values that denote what kind of tile was placed. It is to be used when generating recorded tracks.

- **Truck Recording** - This records the the lane each truck was placed in when it was spawned. It is an array of integers either 1, 2 or 3.
- **Ghost Speed Recording** - This is an array containing the users effective speed (float) for every single physics frame of their previous session.
- **Ghost Lean Recording** - This is an array containing the users effective lean (float) for every single physics frame of their previous session.

The details of how these arrays are built will be discussed in the following sections. The files are written and read from Unity's persistent data path.

5.4 The Ghost and The FeedForward Effect

We have in the previous section mentioned speed and lean values which are stored and used to re-simulate the players behaviour. However, we firstly investigate the implementation specific reasons behind not using a transform based ghost system.

If we use a transform based ghost implementation, a few issues arise. Firstly, the vectors and Euler angles are not serializable. We would have to deconstruct these into serializable primitives such as floats and then reconstruct when loading them back in as saved data. This is not a big problem but it would take a considerable amount of time and effort.

The second more pressing issue is that of interpolation. If we record positions for the ghost only every fixed update (which is what we would have to do because update is not consistently timed as it occurs once every frame), then there will be frames in the game where the ghost is not moving between fixed updates. This will make the ghost look "jittery" and has the potential to be quite jarring unless we create some method of interpolation between values to be used by update instead. In general in Unity, if you want to move rigidbodys smoothly, then it is best done with the physics system (which is what the player currently interacts with).

Outside of the context of our game, a reason you may want to use a position based recording system is if the physics simulations that generated the player movement are very intensive or have an element of randomness. In such cases, recording inputs would not be good since the ghost would also have to be processed through the same physics simulation. This could be intensive or not produce a consistent result.

With regards to our exact input based ghost controller, we essentially reused the player controller but made some adjustments. Instead of the velocities being calculated using speed and lean derived from the hardware input, the speed and lean at every frame are read from a large array containing the full list of speeds and leans for every single frame of the last session. This can be seen in the ghosts FixedUpdate method in Figure 5.11.

Listing 5.11: Using Recorded Inputs to Generate Ghost Movement

```
void FixedUpdate()
{
    if(ghostSpeedRecording != null)
    {
        if(recordingIterator!= ghostSpeedRecording.Length)
        {
            float v = ghostSpeedRecording[recordingIterator];
            float h = ghostLeanRecording[recordingIterator];
            recordingIterator++;
            Move(v, h);
            calculateSeperation();
        }
    }
}
```

This ghost controller FixedUpdate provides the same functionality as the player controllers fixed update. However, the values of **v** and **h** that are used to calculate velocities every frame are not derived from the hardware and are instead accessed from their respective recorded arrays. They are indexed by an iterator that starts at 0 when the ghost is first activated (when the exercise session starts) and is incremented by one every single frame. Since fixed update always occurs 50 times a second, we can be sure that the ghost's simulation will line up perfectly with the users original performance.

These arrays are read in at the beginning of the game from the DataSaver instance as discussed previously. In order to record the ghost, the player controller adds its current speed and lean value to a corresponding list every single physics frame. Once the session is finished, this list is converted into an array. It is important to convert to an array for saving as lists are far less efficient in terms of size and also are marginally slower to access, which could make a difference when it's being accessed every single frame. Increases in efficiency, no matter how small, progress towards fulfilling Requirement 3.3.11.

One problem we encountered with the ghost during implementation was that of getting the ghost on equal footing with the user at the start of every sprint phase. Originally, the ghost and the user would just drive along a track till the end of the session. However, the sense of competition with your ghost was ruined in the sprint sessions if the user was already behind or in front of the ghost. Therefore, we had to find a way to implement the ghost and the player starting in line at the beginning of the sprint phases.

We considered many practical implementations, however each one seemed to come with its own problems:

- **Speeding up/slowing down the player until they're in line with the ghost at the start of the sprint** - whilst this implementation may have looked good and felt natural when the distance between the player and the ghost was small, when the distance separating the two is very large either the player would have to come to a stop or fly along the track at nauseating speeds, both of which would provide significant sensory disconnects between the users cycling action and the actual speed they move at.

- **Speeding up/slowing down the ghost until they're in line with the player at the start of the sprint** - this implementation is similar to the above. However, it alleviates the issues regarding sensory disconnects as we are no longer moving the player and instead moving the ghost. The problem with this, however, is now the players recorded actions (the ghost) do not line up with its position on the track so the ghost may end up driving through trucks, which will ruin the human like behaviour we need the ghost to deliver for the Feedforward Effect to be effective - the user needs to believe they are racing themselves.

- **Teleporting the Player and The Ghost back to the Start** - this solution does not incur any of the issues that the other two have, however teleporting the user could be disorientating and break immersion if not handled suitably.

We decided upon implementing via the teleporting mechanic. In order to not disorientate users with their player view suddenly changing to a different environment, we slowly fade the screen to black, teleport the user and the ghost, then fade the screen back to normal. After this there is a 3 second countdown leading into the sprint where users are fixed in place to ensure they only start moving as the sprint phase starts. The pieces of code that handle the teleporting can be seen in Listing 5.12 and Listing 5.13.

Listing 5.12: The Exercise Protocols Teleport Method

```
void teleport()
{
    //Teleport ghost if there is one
    Vector3 policeSeperator = new Vector3(0.0f, 0.0f,
       (player.transform.position -
       police.transform.position).z);
    GM.restartSprint();
    police.transform.position -= policeSeperator;
    if (GM.ghostExists)
    {
         ghostModel.SetActive(false);
    }
    PC.freezeMovement = true;
}
```

Listing 5.13: The Game Mangers Restart Sprint Method

```
public void restartSprint()
{
    //Destroy everything then commences the initial build
       again.
    while (vehicles[0] == null)
    {
        vehicles.RemoveAt(0);
    }
    while(vehicles.Count != 0)
    {
        removeVehicle();
    }
    while(sections.Count != 0)
    {
        Destroy(sections[0]);
        sections.RemoveAt(0);
    }

    transform.position = Vector3.zero;
    initialBuild();
}
```

Within the initial build method used in Listing 5.13, the player and the ghost are both moved back to the starting positions they are assigned at the start of the game.

With this implementation of the ghost, implementing the Feedforward Effect was really as simple as just increasing the bikes resistance but keeping the ghosts performance consistent. This way the user sees themselves performing at a level that is effectively better than their previous performance, since matching pace takes more effort.

5.5 Track Generation and Recording

When it came to implementation, track generation was a huge part of the game that was ultimately cut out for the final experiment version of the game. Significant work went into creating a procedural generation system that created a track with turns that was far more varied. Similar work went into programming the logic behind the auto navigating of turns for both the player and for the trucks and police cars. We will not investigate the implementation of these items since ultimately they were not used in the final version of the game since the corners were simply too disorientating and made users feel nauseous. However, it is worth mentioning that this was an area we implemented.

Track generation now simply creates the same tile over and over again whenever the users hitbox enters the trigger hitbox on each tile. That is, every time a new tile is entered the game spawns a new tile at the next position and deletes the oldest tile in the list of current tiles .

When the method to spawn a tile is called, it also attempts to delete any trucks that are out of scope (too far behind the player. When a truck is instantiated, a reference to the game object is stored in a list which is used to reference and delete them. The deleting of track and vehicles can can be seen in the code excerpt in Listing **??**. The code checks to see if the oldest vehicle is within range of the section being deleted and, if it is, it deletes it also.

Listing 5.14: Removing the Oldest Track Section and Removing and Vehicles that are Too Far Behind

```
//Removing dead vehicles from the lsit that were killed by
   the police / player.
while (vehicles[0] == null)
{
     vehicles.RemoveAt(0);
}

if (vehicles.Count != 0)
{
     if (Vector3.Distance(vehicles[0].transform.position,
        sections[0].transform.position) < 10.0f)
     {
          removeVehicle();
     }
}
//Removing the first tile in the buffer and destroying the
   gameobject- FIFO system.
Destroy(sections[0]);
sections.RemoveAt(0);
```

When a new section is a created, a new vehicle is also created. In our final implementation, we did not need to record what sections we placed since we always place the same tile. However, the framework exists to record tiles based on an integer representation and store them to the DataSaver instances section list should we want to use the file set of tiles and corner logic again.

Vehicles are handled in a similar way. To ensure that the experience is consistent between sessions, we record what lane each vehicle is spawned in to a list in the DataSaver instance and then in much the same way as the previously described ghost's implementation they are read back out. Every time a new vehicle should be spawned it uses the next value from the recorded list of lane integers.

5.6 Tracking the Exercise Protocol and Changing Intensity States

We used a very simple implementation to define and track our exercise protocol. Within the editor we can simply define a list of float values. This list of floats corresponds to the length of each phase where the length of the list corresponds to number of phases.

Phases are simply high-intensity or low-intensity. as far as the game is concerned there is structurally no difference between a period of low intensity for a warmup, cooldown or recovery as they are all a period of low-intensity.

The script managing the exercise protocol uses the first value in the list and sets a timer to this value when the game begins. The update method decrements the timer by Time.deltaTime which is the time in seconds since the last execution of update (i.e. the time between frames). This continues until the timer value is reduced to 0 and then the nextTime method is called, this code can be seen in Listing 5.15, there is also some other functionality regarding UI text in this method, but for now we are just concerned with the aforementioned parts.

Listing 5.15: Keeping Track of the Time Left in the Exercise Protocol Phase

```
void Update() {
    currentTime -= Time.deltaTime;
    if (currentTime < 5.0f && teleportReady && !intense &&
       timeList.Count != 0)
    {
         teleportReady = false;
         StartCoroutine(FadeToBlack());
    }

    //Makes the time bigger if there is 3 seconds left
    if (preSizeUp && (currentTime < 3.1f))
    {
         //Then gives appropriate prompt based on intensity of
            the phase
         timer.fontSize = 100;
         if (!intense)
         {
              if(timeList.Count != 0)
              {
                   prompt.text = "Ready To Sprint?";
              }
              else
              {
                   prompt.text = "Almost Done";
```

```
            }
        }
        else
        {
            prompt.text = "Ready to Slow?";
        }
        preSizeUp = false;
    }
    timer.text = (currentTime).ToString("F1");

    if (currentTime <= 0 && !gameFinished)
    {
        nextTime();
    }

    //Allowing us to change resistance on the fly with the
       keyboard in training session to help users decide on a
       resistance
    if (Input.GetKeyDown("up"))
    {
        SPC.resistance += 1;
    }
    else if (Input.GetKeyDown("down"))
    {
        SPC.resistance -= 1;
    }
}
```

The nextTime method is important as it is used in the delivery of certain statistics and interface with the intensity controller to change the aesthetic of the scene as well as the current resistance. The nextTime method also triggers the teleport functionality we detailed earlier.

If nextTime is called and there are still values in the list to iterate over, the timer is set to the new timer. Values pertaining to performance are recorded and the intensity state is effectively switched to the opposite state.

If when the nextTime method is called and the list of timers is empty, the nextTime effectively starts a chain of events to end the session including recording total data values we want to get out in the results screen (See Listing 5.16). It also passes lists of timestamps and distances for each phase to the Menu System to be used by the results screen.

Listing 5.16: Calling nextTime with No Phases Remaining in the Defined Exercise Protocol

```
//End Session
gameFinished = true;
prompt.text = "Well Done You've Finished!";
GSM.distances = distances;
GSM.timeStamps = timeStamps;
bool beatGhost = true; //True when there is no ghost set
   false if there is a ghost and the user didn't beat it
   overall.

if (GM.ghostExists)
{
    //Set screen maanger lists
    GSM.ghostDistances = ghostDistances;
    //Calculate total distances covered.
    float playerTotal = 0.0f, ghostTotal = 0.0f;
    for(int i = 0; i < distances.Count; i++)
    {
        playerTotal += distances[i];
        ghostTotal += ghostDistances[i];
    }
    //If user didnt beat ghost set beatGhost to false
    if(playerTotal < ghostTotal)
    {
        beatGhost = false;
    }
}
else
{
    GSM.ghostExists = false;
}
GM.endSession(beatGhost);
```

In this instance, the method will also check to see if the player did not beat their ghost. This occurs when the players total distance is less than the ghosts (assuming there was a ghost in the session). The beatGhost boolean is used when writing the recorded performance to file. If the user beat their ghost, then there is functionality to overwrite the existing recording (although we turned this off for the experiment).

Chapter 6

Evaluation

In this chapter we will detail the methodology of our experiment; specifically our Experimental Design, our Experimental Procedure as well as the Participant Demographic. We will then detail and discuss the results we obtained, as well as evaluation of the experiment.

6.1 Methodology

Aim of the Study - Show that using the feedforward effect with a ghost can improve user performance with negligible loss in intrinsic motivation/enjoyment.

6.1.1 Experimental Design

In this section we discuss the design of the experiment.

The experiment uses a within-subject design meaning that every participant is subject to each of the conditions we intend to investigate. The specific conditions stem from the independent variables we intend to change, and the independent variables we intend to change are:

- **"Ghost"** - **With** the ghost mechanic and **without** the ghost mechanic.
- **"The Feedforward Effect"** - **With** the feedforward effect and **without** the feedforward effect.

Utilizing these independent variables, we created 3 conditions to investigate the effect these variables have on certain dependent variables:

1. **Without** the **Ghost**, **Without** the **Feedforward Effect**
2. **With** the **Ghost**, **Without** the **Feedforward Effect**
3. **With** the **Ghost**, **With** the **Feedforward Effect**

We do not include the fourth combination of "**Without** the **Ghost**, **With** the **Feedforward Effect**", due to the nature of The Feedforward Effect, it extends the functionality of the ghost. It requires a real-time visualization of the user in order to be "The Feedforward Effect", therefore without the ghost we are not utilizing the feedforward effect but instead making the resistance higher with no goal or reason.

It is important to determine the exact specifics of these independent variables:

- **With the Ghost** - the ghost mechanic is enabled in the game, the user has a real-time manifestation of their previous exercise performance.
- **Without the Ghost** - the ghost mechanic is disabled in game, the user has no previous performance to compete with.
- **With the Feedforward Effect** - the resistance is surreptitiously increased by 10% to give the illusion of the users ghost performing at a higher level than it really did, users must work harder to achieve the same performance. This value of 10% has no scientific basis since this is an unexplored application of the Feedforward Effect. However, we wanted to choose a value for which we would observe some statistically significant difference if the mechanic was successful, whilst remaining unnoticeable to the user. 10 percent seemed like a reasonable value to use in this initial experiment.
- **Without the Feedforward Effect** - the resistance is not changed, the work required to perform at the level of the ghost is identical to the original work put in, when it was recorded.

Here we give a detailed breakdown of the users tasks in each of the three conditions:

1. **Without** the **Ghost**, **Without** the **Feedforward Effect** - This is the first condition that forms the basis of comparison with the other two conditions to determine the effectiveness of the ghost and feedforward mechanics. Within this condition users are tasked with playing our game - "Evercycle". They are told to dodge the trucks as they ride along the road, and in the high intensity sections should be at "**near-maximal exertion**", trying to go as far as they can in the sprints. In this session the users inputs are recorded to create the ghost that is used in the subsequent sessions.

2. **With** the **Ghost**, **Without** the **Feedforward Effect** - In this condition, users perform exactly the same tasks playing "Evercycle" again, however, are given the additional task of trying to **Beat their Ghost**. This condition should allow us to determine the effectiveness of the ghost.

3. **With** the **Ghost**, **With** the **Feedforward Effect** - Similarly to the other conditions users are given the same tasks whilst playing our game "Everycycle" and are also given the task to of trying to **Beat their Ghost**. In this condition however the resistance is surreptitiously invreased by 10% making the energy expended in moving the same distance forward 10% greater. This effectively makes the ghost appear to be performing better than the users actual previous performance.

To determine the ordering of these we first considered the requirements of each condition and then the **confounding variables** that could effect the experiment.

Regarding the requirements of the conditions, condition 2 and condition 3 are both being directly compared with condition 1. They require condition 1's performance to be recorded to be then visualized in the sessions as a ghost. Therefore condition 1 must occur first - but this poses problems with regards to potential confounding factors.

Since we are using a within subject design the confounding variables come as a result of of "**carry-over effects**". Here are the two main carry-over effects we **must** consider and the ways we tried to reduce their effect:

- **Practice** - This is the idea that as the user plays through the conditions they will progressively become more confident/better at the game, consequently skewing our results. This is especially important since we intend to compare condition 1 (which must be done first) with condition 2 and 3. If the practice effect occurs then the recorded ghost will not be as good as the later conditions performance, thus potentially inflating our results due to lack of initial skill. However, as the objective of the game is relatively straightforward and the controls are both relatable and intuitive, gaining competency with the game mechanics should take minutes at most. Despite this short time frame, this first carry-over effect could skew our results so we will have a practice session at the start. The session was set before condition 1 to allow users to familiarize themselves with both the format of the game and its controls.

- **Fatigue** - The next issue is that of fatigue. Since this is an exercise game, and specifically a high intensity exercise game, fatigue will play a large part in reducing user performance in subsequent sessions. Since condition 1 always has to be done first this could lead to users over-exerting themselves in the first session and not being able to perform at the same level in the subsequent sessions. This occurred in our pilot study, where the user went at "maximal-effort" for the three sprints back to back (as the standard Wingate Protocol would detail) and consequently could not finish the exercise protocol. To combat this we reduced the number of sprints per session to 2 and increased the recovery time from 30 seconds between sprints to 90 seconds. We also asked participants to cycle at near-maximal effort as to not burn themselves out on the first session. Ultimately a certain amount of fatigue due to sports performance is inevitable, and its effect on certain participants was noted.

Both of these problems derive from order-effects, therefore we should try to mitigate the ordering effect as much as possible by having half the participants carry out condition 2 then condition 3 and the other half vice versa.

This gave us two potential experiment orders that users were randomly assigned (half and half):

- **A** - Condition 1 → Condition 2 → Condition 3
- **B** - Condition 1 → Condition 3 → Condition 2

Now having considered the independent variables and the potential confounding variables within the experiment, we consider the list of dependent variables we will want to measure, as well as why, how, and when we will measure them.

- **Heart Rate** (Measured Consistently Throughout) - We want to keep a graph of heart rate so we can find users peak heart rate throughout each session/sprint for health and safety (don't want to exceed maximum heart rate), this is also often a good representation of users levels of exertion. We will measure this using the Polar H7 heart rate monitor and the Polar Flow fitness application for android.
- **Perceived Exertion** (Measured After Each Sprint) - We want to know how users rate their own levels of exertion as it may be interesting to see how this relates to their actual performance. We will make use the 1 - 10 Borg RPE Scale where 1 is no effort and 10 is maximal effort (See Appendix **??**). We will ask the user for their answer verbally whilst they are still using the game.
- **Performance(Energy)** (After Each Condition) - We want to measure users exercise performance, since this is what we are trying to improve between conditions. We will consider performance to be the **energy** participants expend to rotate the bike pedals (The total energy in Joules that they put into the system). We will investigate the total energy the participant used in only the high intensity sections per condition.
- **Questionnaires** (N/A) - We will utilise an a variety questionnaires to gather quantifiable data on non quantifiable variables such as opinions/emotions/states of mind. We will give the full details of the questionnaires shortly.
- **Qualitative Comments** (After Each Condition) - We will collect qualitative feedback regarding aspects users liked/disliked about the game as this is a good way to determine future work that should be done on the game. We will do this after every condition to see if their opinions / comments change.

- **Perceived Difference** (After the experiment) - We ask users if they noticed any difference between Condition 2 and Condition 3, to see if users noticed the increase in resistance.

Screening Questionnaires

These questionnaires are to be completed and confirmed before potential participants are able to be approved for the study.

- **PAR-Q - The Physical Activity Readiness Questionnaire** - This is a standard screening questionnaire given to individuals who are intending to become much more physically active in the immediate future. The PAR-Q consists of 7 questions concerning the health of the participant. If the participant's answer to any of these is **yes** then they will have to first consult with their doctor before being able to participate in the study (Thomas, Reading and Shephard, 1992).

- **University of Bath Health Departments Health Screen** - We were also required to use a standard health screen frequently used by the University Health Department. In a similar manner to the PAR-Q, if marked questions are answered with yes, a doctors consent is required.

Pre-Experiment Questionnaires

There questionnaires are all to be completed prior to any of the experiment conditions.

- **Demographic Information Questionnaire** - This questionnaire allows us to gather information about the demographic of our participant group. It includes information such as age, gender, occupation, height, weight, BMI, blood pressure, etc. Blood pressure was particularly important to consider since we didn't want to allow participants with blood pressure that was too high or low to take part. It was critical that all of this data was kept anonymously.

- **IPAQ: The International Physical Activity Questionnaire** - This questionnaire generates data based on the levels of physical activity the participant has undergone in the last 7 days. It can be used to establish existing fitness levels or help identify potential reasons for anomalies in data.

- **RM4-FM: Motivation for Physical Activity and Exercise/ Working Out Questionnaires** - This pair of questionnaires investigates the reasons why people are motivated to exercise - Negative numbers reflect that users are extrinsically motivated for change; that is, external factors are important in regulating your behaviour. Whereas, positive numbers reflect that intrinsic motivation is primarily involved in their behaviour. This questionnaire uses a 1 - 7 scale and certain questions hold negative or positive weighting.

- **SOQ: The Sports Orientation Questionnaire** - This questionnaire uses 3 sub-scales:
  - **Competitiveness**: how much do you enjoy competition and strive to succeed? Competition-orientated people enjoy competing and seek out competitions to take part in.
  - **Win Orientation**: how important is winning to you? Win-orientated people compare their performance with other people, rather than setting personal standards.
  - **Goal Orientation**: how important is your own personal performance? Goal-orientated people are "competing against themselves" rather than trying to beat other people

  We believe our ghost implementation is a gamification of challenge/self-competition so it would be interesting to see the relationship between rate of improvement and peoples competitive orientations of participants that use our game.

Post-Condition Questionnaires

These questionnaires are to be answered after each of the three conditions.

- **PPL-FSQ: The Flow State Questionnaire of the Positive Psychology Lab** - This measures two key factors of flow:
  - **Perceived Absorption/Immersion** - We hope to see that if users are absorbed/immersed in the game/task then their performance should reflect this. Ideally the addition of the ghost (condition 2) will increase this and the addition of the feedforward effect (condition 3) wont impact it too significantly.
  - **Balance of Challenge** - In order to be in a state of flow the balance of challenge has to be correct. We will want to see if the state of flow is affected by adding the ghost (condition 2) and how it changes after surreptitiously increasing the resistance (condition 3).
- **IMI: The Intrinsic Motivation Inventory** - This questionnaire can measure up to 7 different sub-scales. We are making use of the following 4:
  - **Intrinsic Motivation - Interest/Enjoyment** - This is the most significant scale in the IMI. We want to measure users motivation and see how it changes between the conditions, so that hopefully we can show that utilisiing the feed-forward effect doesn't reduce motivation.
  - **Effort/Importance** - A subscale of the IMI, it could be interesting to see how users effort and their perceived importance of the activity changes between the conditions.

– **Pressure/Tension** - Another subscale of the IMI, it will be interesting to see how the users sense of pressure/tension changes especially after introducing the ghost.
– **Value/Usefulness** - A subscale of the IMI, it could be interesting to see how the way users value the system changes after racing against their ghost.

Post-Experiment Questionnaires

These questionnaires are to be answered after completing all three of the conditions and post-condition questionnaires.

- **IEQ: Immersive Experience Questionnaire** - This questionnaire is used to quantify how immersive the experience was (Jennett, Cox, Cairns, Dhoparee, Epps, Tijs and Walton, 2008) by calculating each user's IEQ Immersion score. By comparing this to the maximum possible score, we are able to gauge how immersive people thought our virtual reality game was.
- **RM4 - FM: Motivation for Physical Activity and Exercise/ Working Out Questionnaires** - Users initially complete this questionnaire before playing the game and are asked to complete it once again after using playinh. This was used to investigate the general reasons people are motivated to exercise. It would be interesting to see if after playing the game people felt a stronger sense of intrinsic motivation.

6.1.2 Hypotheses

We will investigate 4 hypotheses in this experiment:

- **H1:** Using The Ghost with the Feedforward Effect (increased resistance) will improve exercise performance.

  This is between condition 1 and condition 3, we will determine this based on whether the Total Energy (how we're measuring performance) put into the system is greater across the full set of participants and whether the difference is significant. Ideally we will see the average energy consumed in condition 3 being greater than that used in condition 1 and the difference being statistically significant.
- **H2:** Using the Ghost with the Feedforward Effect (increased resistance) will improve exercise performance by a greater amount than just the Ghost without The Feedforward Effect.

  This is between condition 2 and condition 3, we will measure percentage increase in performance for both conditions in relation to condition 1. We will then compare the average percentage increases to see if condition 3's increase is greater than condition 2's. We will then test to see if the difference between them is of significance. Ideally we will see condition 3 increasing performance by a greater percentage than condition 2 and the difference between them is of statistical significance.

- **H3:** There will be no significant difference in flow and intrinsic motivation/enjoyment when using the Ghost with The Feedforward Effect(increased resistance) and when using it without The Feedforward Effect(increased resistance).
- **H4:** Most people will not notice the increased resistance, when using the Feedforward Effect.

  We will investigate this based on user responses to our perceived difference question. Ideally we would see most people not mentioning the feeling of the resistance being increased in their response.

The Experiment Procedure

Here we clearly layout the full step by step procedure that every participant was run through to complete our experiment.

A foreword; the experiment procedure we ran had to be fully approved by REACH (The Universities - **Research Ethics Approval Committee for Health**), since we were conducting a health study with human participants utilising the University DASH Lab.

This procedure had to be submitted and many elements had to change before our study was approved - most notably regarding the screening process, the risk-related information provided in the Participant Instruction Sheet (See Appendix B) as well as additional criteria that had to be added to our Participant Consent Form (See Appendix C).

Here is the full step-by-step procedure we went through for every participant:

1. Send the potential participant a Google Form Version of the PAR-Q screening questionnaire. Only process participants who "pass" the PAR-Q, to the next stage.
2. Upon passing the PAR-Q, users were told the following pieces of information:
   - Their identification number for the study - to be used to record and generate the ghost in game (a game account effectively).
   - To eat 2 hours before so that participants have suitable levels of energy, but no earlier - since HIIT lead to sickness.
   - To wear comfortable sports clothes for the experiment session (they will be provided private space for changing).
   - Sent a link to general High Intensity Interval Training Information link: https://www.acsm.org/docs/brochures/high-intensityinterval-training.pdf
3. Once the participant arrives at the lab: Ask them to complete the University's standard Health Screen Questionnaire and ask them if they're feeling okay to participate, otherwise abort the experiment.

4. Ask the participant to read through the full participant instruction sheet (See Appendix B).
5. Ask the participant to read through and subsequently sign the Participant Consent Form (See Appendix C).
6. Ask participant to get changed but leave their shoes off for now and put on Polar HR monitor chest strap, explaining how to do so. Participants are given privacy in a side-room of the lab. Offer water as refreshment
7. Take body related measurements for the Demographic Questionnaire:
   - Height
   - Weight
   - Blood Pressure - if this is too high (140/90) or low (90/60) abort the study.
   - Body Composition
   - Calculate BMI
8. Calculate resistance to use during sprints based on body weight: 0.40 Nm $\times$ Kg-1
9. Ask the user to complete all of pre-experiment questionnaires.
10. Show them 10-point Borg RPE scale and tell them that you will ask them to rate their RPE as felt during the sprint immediately after every sprint.
11. Reiterate the structure of the game, also tell users that they should be sprinting in the high-intensity sections (but make sure they're aware they will have to sprint 6 lots of 30 seconds).
12. Calibrate all the appropriate hardware:
    - Set the exercise bike seat height to an appropriate comfortable level.
    - Fit the HTC Vive HMD to the users head and make sure it is well secured and comfortable.
    - Ask the participant to take their cycling position in order to calibrate the center position of the HMD.
13. Begin the training phase roughly 3-4 minutes - allow users to get used to the controls and the different phases. It is at this stage we should ask users if they want to raise or lower the resistance based on their skill level to find the correct resistance such that the users can cycle fast without their legs spinning out of control. The users should complete some sprint sections within the training session to make sure they are comfortable with the system.
14. Ask users to give a rating for Rated Perceived Exertion for a sprint in the training session so they are familiar with the process.

15. Ask participants whether they have understood the procedure and whether they have any questions. Answer questions if necessary. Should result in 1 minute break.
16. Start Condition 1, recording the player's inputs to create the ghost to be used in Condition 2 and 3. Within this condition:
    - Make sure Participant number and HIIT parameters are correctly entered.
    - Ask for 1-10 RPE after every sprint.
17. Ask participant to complete the post-condition questionnaires, offer them water to keep them hydrated.
18. Start Condition 2 (or Condition 3 if counterbalanced - preventing the order-effect). Within the condition:
    - Make sure Participant number and HIIT parameters are correctly entered.
    - Ask for 1-10 RPE after every sprint.
19. Ask participant to complete the post-condition questionnaires, offer them water to keep them hydrated.
20. Start Condition 3 (or Condition 2 if counterbalanced - preventing the order-effect). Within the condition:
    - Make sure Participant number and HIIT parameters are correctly entered.
    - Ask for 1-10 RPE after every sprint.
21. Ask participant to complete the post-condition questionnaires, offer them water to keep them hydrated.
22. Ask the participant to complete the post-experiment questionnaires.
23. Debrief participants and answer any questions they may have.

Here are the various **stopping criteria** we had in place for our experiment procedure:

1. Suspicion of a myocardial infarction or acute myocardial infarction (heart attack).
2. Onset of moderate-to-severe angina (chest pain).
3. Signs of poor perfusion (circulation or blood flow), including pallor (pale appearance to the skin), cyanosis (bluish discoloration), or cold and clammy skin.
4. Severe or unusual shortness of breath.
5. CNS (central nervous system) symptoms e.g., ataxia (failure of muscular coordination), vertigo (An illusion of dizzying movement), visual problems, confusion.
6. Patient's request (to stop).
7. Severe fatigue.

6.1.3 The Participant Demographic

From our participants demographic questionnaires we determined the following statistics about the participants that took part in our study:

All values are rounded to two decimal places.

- Body Height ranged from 162.3cm to 194.3cm (mean=180.31cm, SD=9.63).
- Body Weight ranged from 57.7 kilograms to 97.8 kilograms (mean=75.23KG, SD=15.48).
- Body Composition ranged from 7.1 to 28.7 (mean=18.24, SD=8.17).
- BMI ranged from 18.38 to 31.91 (mean=23.10, SD=4.23).
- Average hours spent exercising per week ranged from 1 to 12 (mean=5.375, SD=3.74).
- Average hours spent playing video games per week ranged from 0 hours to 15 hours (mean=3.63 hours, SD=5.733).
- Two participants required glasses due to short-sightedness.
- One participant was wearing contact lenses for short-sightedness.
- No participants reported any form of colour-blindness.

From the IPAQ we determined the following physical activity in the last 7 days from our participants:

- Seven participants participated in some form of vigorous physical activity in the past 7 days
- One participant reported 5 days of vigorous physical activity, one reported 3 days, three reported 3 days, two reported 2 days and one reported 1 day.
- The time per day spent performing vigorous physical activities ranged from 10 minutes to 2 hours (mean=49 minutes, SD=0.71).
- Six participants reported some form of moderate physical activity.
- The time per day spent performing moderate physical activities ranged from 20 minutes to 90 minutes (mean=38 minutes, SD=0.51).
- All participants except one walked for at least 10 minutes every single day, the other participant walked 10 minutes for 5 out 7 days.
- The time per day spent walking ranged from 20 minutes to 3 hours (mean=62 minutes, SD=0.84).
- Time spent sitting down ranged from 6 hours per day to 12 hours per day (mean=7.85 hours, SD=1.9).

6.2 Results

Here we provide for the results of our experiments, which will drawn upon to support our hypotheses. There was a huge amount of data gathered in this study across just 8 participants, so only the data relevant to proving my hypotheses and other interesting related data will be included in this section.

The full raw data that all the graphs and tables were created with can be found in the appendix.

6.2.1 Performance

Firstly we have the total energy used in the sprint sections across each of the three different conditions. We use Energy (Joules) to quantify performance. To calculate energy we work backwards from the power equation established previously when determining the bike protocol.

Since our sprint sections total to 60 seconds we know that the total energy put into the bike system by the user over a single condition is equal to:

$$TotalEnergy(J) = 60 \times AveragePower(Js^{-1})$$

This is because the power at any moment in time is in Joules per Second. If we can determine the average power level then we can calculate the exact amount of energy the user put into the system.

Our game records the total revolutions over the sprint sections for each condition. We can use the following equation to obtain the RPM we need to determine average power:

$$RPM = TotalRevolutions \times \frac{TotalLengthofSprints(s)}{60seconds}$$

From the calculated RPM and the previously found resistance, we can calculate the average power:

$$AveragePower(Js^{-1}) = Resistance(Nm) \times RPM \times constant$$

The value we refer to as constant was established via trial and error when reverse engineering the setting of resistance. Its value will depend on the length of the flywheel, but the exact relationship between these two was disregarded, as we only needed to establish that it takes a value of 0.11 in our implementation. We then use Average Power in the initial equation to determine the Total Energy used.

This is how we determined the results for Performance as seen in Table 6.1.

Table 6.1: Performance - The Total Power used By Each User across Each Condition

Participant Number	Condition 1 - Total Power (J)	Condition 2 - Total Power (J)	Condition 3 - Total Power (J)
101	16250.058	16715.16	19712.88
104	27408.216	29789.1	30677.856
102	21089.574	22415.976	25987.764
106	19359.648	22828.344	24447.654
108	25074.456	26788.872	28699.044
109	21956.22	25345.98	25892.064
103	13097.04	13679.16	14855.412
105	21316.68	21396.144	22882.464
Mean Average	20693.987	22369.842	24144.392
Standard Deviation	4561.48	5238.515	5035.464

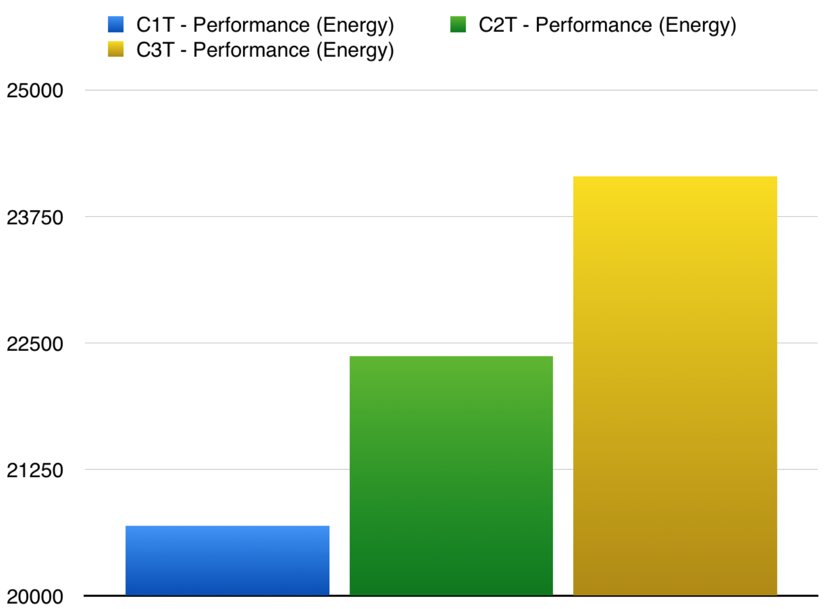


Figure 6.1: A Graph Illustrating the Average Total Energy Used in Each Condition

.

Considering Hypothesis 1: from our results it appears that exercise performance increased significanty from condition 1 to condition 3 (See Figure 6.1). The mean average percentage increase in performance is **16.99%**.

However we should test to see if the difference is of actual statistical significance. The Wilcoxon Signed-Rank Test is a non-parametric statistical hypothesis test that can be used when comparing repeated measures on a single sample. This test is appropriate for our data since we are using the same group of participants for all our conditions. We will use this to establish whether or not the difference in population means is statistically significant. Using the Wilcoxon Signed-Rank Test with a signifiance level p of 0.01 with a twin-tailed hypothesis will provide us with good evidence as to the statistical significance of the sample.

Using the Wilcoxon Signed-Rank Test on Condition 1 and Condition 3's performance values (Energy in joules) with a significance level p of 0.01 and a twin-tailed hypothesis, gives us a W-value (we use the W-value rather than the Z-value since our sample size is too small to utilise the Z-value) of 0. The critical value for a sample size of 8 at $p \leq 0.01$ is 1. Therefore we can disprove the corresponding null hypothesis, and suggest that there is strong evidence that this increase in Performance is significant and not simply a coincidence.

This gives us some evidence to suggest that Hypothesis H1 may be correct, and that the use of the ghost in conjunction with feedforward effect may well improve user performance.

Considering Hypothesis 2: Similarly we can see from Figure 6.1 that Condition 3's average performance was higher than that of Condition 2's. However, to determine the validity of our hypothesis we need to consider percentage increases in performance relative to Condition 1 for both Condition 2 and Condition 3.

The full table of performance increase percentages can be seen in Table 6.2.

Table 6.2: Performance - The Performance Increase of Condition 2 and 3 in Comparison with respect to Condition 1

Participant Number	Condition 2 - Performance Increase (%)	Condition 3 - Performance Increase (%)
101	2.862155938	21.30959779
104	8.686752907	11.92941562
102	6.289373128	23.22564695
106	17.91714395	26.28150057
108	6.837300877	14.45530065
109	15.43872306	17.92587249
103	4.444668414	13.42572062
105	0.3727785002	7.345346461
Mean Average	7.856112096	16.98730014
Standard Deviation	6.041295867	6.352813148

Once again we can see that the average performance increase in Condition 3 with respect to Condition 1, is over double that of the performance increase in Condition 2 with respect to Condition 1. However, we should test using the Wilcoxon Signed-Rank Test, to determine whether or not the increase is part of a trend or simply coincidence.

Using a significance level p of 0.01 with a twin tailed-hypothesis we obtain a W-value of 0 where the Critical Value is 1 for a sample size of 8. This once again rejects the corresponding null hypothesis and provides evidence that this difference is not simply a coincidence and that there may be a relationship between the amount performance is increased by and the use of The Feedforward Effect. We have found strong evidence to in support of hypothesis H2.

6.2.2 Flow and Intrinsic Motivation/Enjoyment

To investigate how the addition of the feedforward effect affected flow and intrinsic motivation/enjoyment we must look at both of the PPL-FSQ subscales for Perceived Absorption/Immersion and Skill/Balance. We should then also the IMI subscale for Intrinsic Motivation - Interest/Enjoyment, to determine the effect the condition has had on enjoyment.

First we will consider the PPL-FSQ scores. This score is generated by first reversing answers that are "negative-scorers" - these scores are set to be 6 minus the score, such that 5 becomes 1, 4 becomes 2 and so on.

After doing this and calculating averages, we came to the final scores for Perceived Absorption/Immersion, Skill/Balance and finally the total Flow Score as seen in the table below. The first 10 questions on the questionnaire contribute to the value for Perceived Absorption whilst the subsequent 9 contribute to Skill/Balance. All the questions contribute to the full Flow Score.

The average values with regards to flow, across all three conditions, can be seen in Table 6.3. Between Condition 2 and 3 there seems to be very negligible difference in the average flow score.

Table 6.3: Average Flow Values Across all 3 Conditions

	Condition 1	Condition 2	Condition 3
Perceived Absorption/Immersion (1-5)	4.333	4.267	4.125
Skill/Balance (1-5)	4.296	4.377	4.347
Flow Score (1-5)	4.315	4.321	4.229

Once again we will use every participants paired flow score between Conditions 2 and 3 with the Wilcoxon Signed-Rank Test to determine whether or not there is a statistically significant difference between them. The standard null hypothesis of this test is that there is effectively no statistical difference between the conditions mean results across the same population. We use a significance level p of 0.05 (allowing for wider leniency) with a twin-tailed hypothesis to solidify the evidence that flow score has not been significantly affected when using the feedforward effect. This gives us a **W**-value of 16.5 where the critical value is 3 when the significance level $p \leq 0.05$ and the sample size in 8.

Since we cannot disprove the null hypothesis there is no evidence to suggest that the use of the Feedforward Effect has any effect on the state of flow the user experiences playing the game. Their average values are also very similar.

Next we will similarly assess the IMI's Enjoyment subscale, this uses a 1 to 7 scale where

1 means not true and 7 means very true. In similar fashion to the PPL- FSQ scale certain answers that are considered "negative-answers" have their scores reversed in a the same way. After this is done we calculate an average score for this subscale across conditions and then see if there is significant difference to disprove the standard null hypothesis. If there is not, then we can infer that using the Feedforward Effect does not have an impact on participants levels of enjoyment and thusly does not reduce enjoyment.

The average user enjoyment scores across all three conditions can be seen in Table 6.4.

Table 6.4: Enjoyment Levels Across all 3 Conditions

UserID	Condition 1 - Enjoyment (1-7)	Condition 2 - Enjoyment (1-7)	Condition 3 - Enjoyment (1-7)
101	5.85714285714286	6.71428571428571	6.71428571428571
103	6.71428571428571	6.71428571428571	6.14285714285714
104	5.14285714285714	5.85714285714286	6.71428571428571
102	6.42857142857143	6.42857142857143	6.28571428571429
105	4.71428571428571	4.28571428571429	6.14285714285714
109	6.57142857142857	6.57142857142857	5.42857142857143
106	5.85714285714286	5.85714285714286	6.42857142857143
108	5.85714285714286	6.57142857142857	5.14285714285714
Average	5.89285714285714	6.125	6.125

The average values for condition 2 and condition 3 are identical so if we can also prove that the Wilcoxon Signed Rank Tests null hypothesis remains true for this data we can assume there is no evidence to suggest that the Feedforward Effect has any affect on participants enjoyment.

We use a significant level p of 0.05 (allowing for wider leniency again) with a twin-tailed hypothesis to solidify the evidence that enjoyment has not been significantly affected when using the feedforward effect. This gives us a **W**-value of 16.5 where the critical value is 3 when the significance level $p \leq 0.05$ and the sample size in 8.

We have not disproved the null hypothesis and there is no evidence to contradict hypothesis H3. Thus we can assume that H3 remains true and that as far as we can infer, the Feedforward Effect has no effect on participants levels of enjoyment or state of flow.

6.2.3 Perceived Difference

Unfortunately we have no way of quantifying user responses and nothing to compare this against so we will simply look at the percentage of participants that noticed the added resistance in condition 3, and give a rough basis as to whether or not we can disprove H4.

The users "Perceived Difference" responses were as follows:

1. "Yes, the ghost pushed me to peddle faster"

2. "No"
3. "I think the ghost in the final round was definitely harder... I felt like I had to work much harder to pull away from the ghost!!"
4. "No"
5. "I found the 3rd session more difficult and straining on my legs"
6. "I was perhaps a little anxious about falling behind the ghost. Instead of trying to go as fast as possible, I guess I was trying to maximise the distance ahead of the ghost rather than overall speed."
7. "I felt that the resistance was slightly harder. I also noticed that the trucks on the road had parted and left a gap in the center to sprint through. Other than that, I did not perceive any differences."
8. "I found I was far more driven to push my limits when chasing my ghost. I knew that it was possible to cycle as fast as my ghost, so I knew it was possible to beat it. In the first condition I tried fairly hard but within my own limits; in the other two I was completely determined to beat my ghost no matter what."

Of these answers only 3 and 7 pull out the concept of the resistance feeling higher, however answer 5 said they found the third session more difficult (this was the one with the increased resistance), so whilst not explicit it appears they did notice it's affect.

Whilst the majority of people did not notice, and we could technically say that hypothesis H4 remains true, what we really wanted was this resistance increase to be completely unnoticeable. So we will conclude that hypothesis H4 is not completely true.

6.3 Discussion

Through statistical analysis we managed to provide significant evidence to suggest that the statements made in our hypotheses were true. We will discuss the reasons we think we obtained these results in this section.

6.3.1 H1: Using The Ghost with the Feedforward Effect (increased resistance) will improve exercise performance.

Our performance results overwhelmingly support this claim, the difference was proved to be statistically significant even when using an incredibly precise significance level. However, when investigating peak heart rates, the change was very minimal in comparison. We should consider what other factors in the experiment could have lead to the results we saw regarding hypothesis H1.

The first potential issue could be that determining performance as energy put into the bike system is not fully representative of all the components that make up exercise performance. This could potentially explain why the peak heart rate values didn't see as significant a rise as the energy levels did. During exercise there will undoubtedly be energy exerted in other ways that are not directly into the exercise bike system. If we were to investigate this effect again in the future it would be of use to have multiple measures of performance in place.

Whilst there are these potential issues, the evidence that the Feedforward Effect, in conjunction with a ghost, can improve user performance is significant. Providing evidence to suggest that the Feedforward Effect is a viable concept to gamify in order to help people improve at exercise and in turn establish The Feedforward Effect as an effective technique for exercise gamification.

6.3.2 H2: Using the Ghost with the Feedforward Effect (increased resistance) will improve exercise performance by a greater amount than just the Ghost without The Feedforward Effect.

Similarly to H1, we saw overwhelming evidence to support this hypothesis. It was unsurprising to see that the ghost mechanic helped users improve (since this had already been extensively studied). When using the ghost with the Feedforward Effect participants effective exercise improvement doubled, which is an incredibly significant amount.

Between the two conditions there weren't really any other factors that could have contributed to providing such strong support for Hypothesis 2, aside from what was already mentioned with regards to hypothesis H1. We have good evidence to suggest that building the Feedforward Effect into a virtual reality exercise game is more effective at helping users improve than simply utilising a ghost on it's own.

6.3.3 H3: There will be no significant difference in flow and intrinsic motivation/enjoyment when using the Ghost with The Feedforward Effect(increased resistance) and when using it without The Feedforward Effect(increased resistance).

We saw very marginal differences in overall flow scores and user enjoyment. Considering 5 out of our 8 participants didn't even notice a change between condition 2 and 3 it is not surprising that these scores didn't really change.

That being said there is the possibility that whilst the flow and enjoyment average remained relatively consistent, some participants may have increased their flow score and enjoyment levels whilst others may have simultaneously reduced theirs.

There is potential for a very polarizing shift in answers that could have lead to a consistent average. Perhaps the increase in difficulty made users who enjoy competition have more fun, whilst those who are less competitively focused had less fun playing the game with the

Feedforward Effect incorporated.

These possibilities considered, we should not discredit what we have found here. We have not only proved that the Feedforward Effect in conjunction with a ghost of players previous performance is a good way to motivate and help users improve, we've also shown that The Feedforward Effect does not seem to have a significant impact on the quality of the user experience.

6.3.4 H4:Most people will not notice the increased resistance, when using the Feedforward Effect.

We didn't have a good way of statistically proving this claim either way. Almost half the participants mentioned something akin to rise in resistance which is ultimately far more than we'd of liked. Interestingly though this didn't seem to affect the participants enjoyment of the game or even shift users out of a state of flow.

Perhaps users were not shifted out of flow because prior to the increased resistance the game was simply not challenging enough. If the game was already at the perfect level of skill, we'd at least expect the users who identified the game getting harder having a reduced flow score. Regardless it seems like there is a fine balancing point with which the Feedforward Effect can be used to help people improve at physical exercise without them noticing.

A large portion of our participants had no idea that the resistance was increased, however, others had. Interestingly it seemed to be the two users using the greatest resistance that noticed the difference in resistance between condition 2 and 3. This raises the idea that perhaps it's flat resistance increase that is more noticeable rather than just the percentage change.

We would like to conduct further study into what the optimal value to increase the resistance by is when implementing the Feedforward Effect in a virtual reality exercise game. This point of balance certainly exists but this study made no investigation to determine what that point is.

6.3.5 Experiment Limitations

Most of the limitations of the experiment as well as the evaluation of results came down to time. Since getting REACH approval took longer than we anticipated we were only left with a short amount of time to conduct our studies and evaluate results. As a result of this there were immediate limitations.

Firstly the sample size we used was too small, meaning there is a very large variance and which makes it difficult to be sure that any statistical analysis we carried out holds true weight. It follows that making assumptions about the distribution of our data with such a low number of participants may not result in an accurate representation of results. With a small sample size we are unable to conduct more extensive statistical analysis using statistical analysis software such as SPS.

The time constraints also lead to a limited scope as to what we could evaluate in our study. We had a large amount of data via our questionnaires but unfortuantely not enough time to investigate all of it. Although this can't be helped, in the future we would like to fully investigate the wide array of independent variables we recorded.

6.3.6 Participants Opinions on our game - "Evercycle"

There were a large amount of positive comments regarding our game even before playing with the ghost:

- "I really enjoyed it"
- "Thought it was fun and very motivating"
- "Really enjoyed this first run - simple to use, and I actually had fun doing cardio!"
- "I enjoyed the experience but found it challenging. a lot of fun, intuitive, engaging."
- "It was exciting and interesting, and I found it to be a fun challenge."
- "I think this could definitely be used to make exercise more easy for people who find exercise challenging. I felt quite hot inside the headset"

After playing with the ghost many comments mentioned an increased sense of motivation, which didn't seem to change when the Feedforward Effect was added:

- "Very engaging, pushed me to better than I would have done without VR."
- "Very good VR experience and an incredible way to exercise! Really fun racing the ghost!"
- "I liked the challenge of racing my ghost and dodging the vans was fun was too. Would have been nice to be able to see my heart rate"
- "I continued to find the game engaging and fun. It was motivating to see my ghost in front of me and made me want to push myself more."
- "I liked that the ghost provided a significant amount of motivation to beat my previous goal."
- "I was driven to push myself more than I otherwise would have been. I found the challenge of catching up to my ghost to match my skill level well"

Chapter 7

Conclusions

In this section we start by outlining the contributions that our project and software made to the field. Drawing on the potential of our finished project we discuss options for it's use in the future, from an academic and non-academic viewpoint. We conclude this section with some personal reflections regarding the entire project and the process of getting to this conclusion.

7.1 Contribution

Here we outline the key areas of contribution that our project has made to field.

7.1.1 A Comprehensive Evaluation of the Needs of People that Struggle to Maintain/ Improve at Exercise.

Through extensive literature review we determined the exact needs of our systems end users. In order to deliver effective gamification of exercise we identified the exact needs of people who struggle to maintain/improve at exercise. We detailed the lack of self-regulatory skills that are crucial in maintaining and improving at exercise. We also highlighted the key barriers to traditional exercise that prevents users from making the initial transition into a stage of maintained exercise. We determined the exact motivational factors that are found in people who regularly exercise, to identify how these could be utilised in a virtual reality exercise game.

7.1.2 Identified Suitable Game Mechanics that can Help People Improve at Exercise.

Building on these identified motivational factors, we highlighted the most effective game mechanics to deliver the key skills needed to improve at exercise to users that lack these skills. This includes game mechanics that alleviate the need for users to possess self-regulatory skills, that are key to exercise improvement and maintenance.

7.1.3 Effective Gamification of Challenge/Self-Competition via a Ghost.

Focusing on the concept of gamification of self-competition we investigated the effectiveness of ghosts and implemented our own ghost in a virtual reality exercise game. We discussed the nuances surrounding designing ghost game mechanic and highlight key design decisions that other designers wishing to create a virtual reality exercise ghost should consider.

7.1.4 Extended the Functionality of Ghosts using the Feedforward Effect.

We investigated the effectiveness of the Feedforward Effect in other areas and considered the potential ways it can be translated to a video game environment to enable users to perform at levels higher than they're usually capable. We propose the ways the Feedforward Effect can be utilised

7.1.5 Conducted an Empirical Study To Evaluate the Effectiveness of our Feedforward Ghost

We carried out an official Health Department approved exercise related study into the effectiveness of our Feedforward Effect implementation in a virtual reality exercise game. We found sufficient evidence to show that when used in conjunction with a ghost the Feedforward effect can significantly increase exercise performance (more so than just a traditional ghost), whilst not reducing user enjoyment or disrupting the users state of flow. We have shown that the Feedforward Effect can be applied with ease to a video game environment and hopefully this will serve as a basis for future usage of the Feedforward Effect in the gamification of exercise.

7.2 Future Work

It is clear that the Feedforward Effect has a very practical future ahead of it in virtual reality exercise game design. It lends itself very well the video game format and is a wholly unexplored area of gamification. In the future we would like to extend the feedforward effect to other fields of gamification - perhaps identifying how it can be applied to education of business.

Regarding this study there is still potential evaluation that could be carried on, many of the dependent variables we gathered data on we were unable to make full use of simply due to time constraints. We should further investigate how users levels of motivation towards exercise differed from the pre-experiment RM4-FM questionnaire and the post-experiment questionnaire. Similarly we should evaluate the potential trends between the amount exercise performance increased by and the individuals competitiveness as represented by subscales of the Sports Orientation Questionnaire we used.

In terms of the future of our project there is still a huge amount that can be done with it. Firstly, we should conduct further study into identification of the resistance sweet spot, finding the exact percentage or even flat increase that makes the effect undetectable would be perfect.

We should not dismiss the actual desired use of the game either, conducting studies is interesting but what we have created here could also have some functional value to a lot of people. For example the game could be used to potentially accelerate rehabilitation, it could also potentially be refactored for use with a handcrank to help people who can't use their legs improve their exercise performance. The game has potential for use in a more professional sporting environment also, allowing athletes to compete with themselves in real-time is a relatively unexplored concept.

The game also has commercial potential, recently we posted a video of it online and it gained significant media traction (50,000 views in a few days), with a huge amount of the general public being displaying interest in using a similar system. An article was also written about the game in Yahoo Finance, and hopefully this will help with the accelerated development of Virtual Reality Exercise games in general.

7.3 Reflection

I am certainly proud of the virtual reality game I created, it was an incredibly large scale project and I learnt a huge amount in creating. The total project was thousands of lines of code long and hundreds of man hours of detailing. However, in reflection I wish I'd brought the scope of the project in a bit, it became very obvious very quickly that the creating the game was dominating all of my time and ultimately that was not the focus of the final year project. I could have investigated the same effect (which I genuinely found fascinating) with a simpler implementation.

In retrospect it was great fun to take on an enormous project but for the sake of my studies it may have been better to focus more on the dissertation side of the project as much as I focused on creating the actual game. Running out of time due to getting approved by REACH so close to the deadline, made me feel like I couldn't fully utilise and evaluate all the data I collected. There is a very interesting concept at work regarding the use of the Feedforward Effect and I hope Christof and his team continue to investigate it's effectiveness.

Overall, I am overwhelmingly happy with the literature review that was carried out, the design of the project and the implementation. Unfortunately, I feel like there was more evaluation that could have been done using the data we collected, but ultimately we managed to show the potential that the Feedforward Effect has in exercise games and I believe we created a unique and interesting piece of software.

Bibliography

Baker, C. (2017), 'Obesity statistics', *House of Commons: Briefing Paper 3336* .

Ball, K., Crawford, D. and Owen, N. (2000), 'Obesity as a barrier to physical activity', *Australian and New Zealand journal of public health* **24**(3), 331–333.

Banos, R. M., Escobar, P., Cebolla, A., Guixeres, J., Alvarez Pitti, J., Lisón, J. F. and Botella, C. (2016), 'Using virtual reality to distract overweight children from bodily sensations during exercise', *Cyberpsychology, Behavior, and Social Networking* **19**(2), 115–119.

Bolton, J., Lambert, M., Lirette, D. and Unsworth, B. (2014), Paperdude: a virtual reality cycling exergame, *in* 'CHI'14 Extended Abstracts on Human Factors in Computing Systems', ACM, pp. 475–478.

Burguillo, J. C. (2010), 'Using game theory and competition-based learning to stimulate student motivation and performance', *Computers & Education* **55**(2), 566–575.

Charles, D., Charles, T., McNeill, M., Bustard, D. and Black, M. (2011), 'Game-based feedback for educational multi-user virtual environments', *British Journal of Educational Technology* **42**(4), 638–654.

Cornick, J. E. and Blascovich, J. (2015), 'Virtual reality and eating, diabetes, and obesity', *The Digital Patient: Advancing Healthcare, Research, and Education* pp. 181–197.

Coulson, S. E., Adams, R. D., O'Dwyer, N. J. and Croxson, G. R. (2006), 'Physiotherapy rehabilitation of the smile after long-term facial nerve palsy using video self-modeling and implementation intentions', *Otolaryngology–Head and Neck Surgery* **134**(1), 48–55.

Cropley, M., Ayers, S. and Nokes, L. (2003), 'People don't exercise because they can't think of reasons to exercise: an examination of causal reasoning within the transtheoretical model', *Psychology, health & medicine* **8**(4), 409–414.

Csikszentmihalyi, M. (2014), *Toward a psychology of optimal experience*, Springer.

Cugelman, B. (2013), 'Gamification: what it is and why it matters to digital health behavior change developers', *JMIR Serious Games* **1**(1), e3.

Dale, S. (2014), 'Gamification: Making work fun, or making fun of work?', *Business Information Review* **31**(2), 82–90.

Darby, K. (2014), 'Our encore: running from the zombie 2.0', *Studies in Theatre and Performance* **34**(3), 229–235.

Deterding, S., Dixon, D., Khaled, R. and Nacke, L. (2011), From game design elements to gamefulness: defining gamification, *in* 'Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments', ACM, pp. 9–15.

Dishman, R. K. (1991), 'Increasing and maintaining exercise and physical activity', *Behavior Therapy* **22**(3), 345–378.

Dowrick, P. W. (2012*a*), 'Self model theory: Learning from the future', *Wiley Interdisciplinary Reviews: Cognitive Science* **3**(2), 215–230.

Dowrick, P. W. (2012*b*), 'Self modeling: Expanding the theories of learning', *Psychology in the Schools* **49**(1), 30–41.

Fabricatore, C. (2007), 'Gameplay and game mechanics: a key to quality in videogames'.

Gears, D. and Braun, K. (2013), Gamification in business: Designing motivating solutions to problem situations, *in* 'Proceedings of the CHI 2013 Gamification Workshop'.

Gibala, M. J., Little, J. P., MacDonald, M. J. and Hawley, J. A. (2012), 'Physiological adaptations to low-volume, high-intensity interval training in health and disease', *The Journal of physiology* **590**(5), 1077–1084.

Gibala, M. J. and McGee, S. L. (2008), 'Metabolic adaptations to short-term high-intensity interval training: a little pain for a lot of gain?', *Exercise and sport sciences reviews* **36**(2), 58–63.

Hagin, V., Gonzales, B. R. and Groslambert, A. (2015), 'Effects of cognitive stimulation with a self-modeling video on time to exhaustion while running at maximal aerobic velocity: a pilot study', *Perceptual and motor skills* **120**(2), 491–501.

Hamari, J., Koivisto, J. and Sarsa, H. (2014), Does gamification work?–a literature review of empirical studies on gamification, *in* '2014 47th Hawaii International Conference on System Sciences', IEEE, pp. 3025–3034.

Herzig, P., Strahringer, S. and Ameling, M. (2012), Gamification of erp systems-exploring gamification effects on user acceptance constructs, *in* 'Multikonferenz Wirtschaftsinformatik', GITO Braunschweig, pp. 793–804.

Huang, W. H.-Y. and Soman, D. (2013), 'Gamification of education', *Research Report Series: Behavioural Economics in Action, Rotman School of Management, University of Toronto* .

IJsselsteijn, W. A., Kort, Y. d., Westerink, J., Jager, M. d. and Bonants, R. (2006), 'Virtual fitness: stimulating exercise behavior through media technology', *Presence: Teleoperators and Virtual Environments* **15**(6), 688–698.

Ingledew, D. K., Markland, D. and Medley, A. R. (1998), 'Exercise motives and stages of change', *Journal of health psychology* **3**(4), 477–489.

Iso-Ahola, S. E. and Clair, B. S. (2000), 'Toward a theory of exercise motivation', *Quest* **52**(2), 131–147.

Jennett, C., Cox, A. L., Cairns, P., Dhoparee, S., Epps, A., Tijs, T. and Walton, A. (2008), 'Measuring and defining the experience of immersion in games', *International journal of human-computer studies* **66**(9), 641–661.

Kolasinski, E. M. (1995), Simulator sickness in virtual environments., Technical report, DTIC Document.

Kumar, J. (2013), Gamification at work: Designing engaging business software, *in* 'International Conference of Design, User Experience, and Usability', Springer, pp. 528–537.

Laursen, P. B. and Jenkins, D. G. (2002), 'The scientific basis for high-intensity interval training', *Sports Medicine* **32**(1), 53–73.

LeBlanc, A. G. and Chaput, J.-P. (2016), 'Pokémon go: A game changer for the physical inactivity crisis?', *Preventive Medicine* .

Li, B., Maxwell, M., Leightley, D., Lindsay, A., Johnson, W. and Ruck, A. (2014), Development of exergame-based virtual trainer for physical therapy using kinect, *in* 'Games for Health 2014', Springer, pp. 79–88.

Lister, C., West, J. H., Cannon, B., Sax, T. and Brodegard, D. (2014), 'Just a fad? gamification in health and fitness apps', *JMIR serious games* **2**(2), e9.

Masters, K. S. and Ogles, B. M. (1998), 'Associative and dissociative cognitive strategies in exercise and running: 20 years later, what do we know?', *The Sport Psychologist* **12**(3), 253–270.

Morgan, W. P. and Pollock, M. L. (1977), 'Psychologic characterization of the elite distance runner', *Annals of the New York Academy of Sciences* **301**(1), 382–403.

Niedenthal, S. (2009), 'What we talk about when we talk about game aesthetics'.

Norris, S. R. and Petersen, S. R. (1998), 'Effects of endurance training on transient oxygen uptake responses in cyclists', *Journal of Sports Sciences* **16**(8), 733–738.

Olsen, O., Sjøhaug, M., Van Beekvelt, M. and Mork, P. J. (2012), 'The effect of warm-up and cool-down exercise on delayed onset muscle soreness in the quadriceps muscle: a randomized controlled trial', *Journal of human kinetics* **35**(1), 59–68.

Raczkowski, F. (2013), 'Its all fun and games a history of ideas concerning gamification', *DiGRA 2013: DeFragging Game Studies* .

Redfern, M. S., Yardley, L. and Bronstein, A. M. (2001), 'Visual influences on balance', *Journal of anxiety disorders* **15**(1), 81–94.

Shaw, L. A. (2014), Development and evaluation of an exercycle game using immersive technologies, PhD thesis, The University of Auckland New Zealand.

Shaw, L. A., Buckley, J., Corballis, P. M., Lutteroth, C. and Wuensche, B. C. (2016), 'Competition and cooperation with virtual players in an exergame', *PeerJ Computer Science* **2**, e92.

Shaw, L. A., Wnsche, B. C., Lutteroth, C., Marks, S. and Callies, R. (2015), Challenges in virtual reality exergame design, *in* 'Proceedings of the 16th Australasian User Interface Conference (AUIC)', Australian Computer Society, pp. 61–68.

Song, H., Kim, J., Tenzek, K. E. and Lee, K. M. (2010), The effects of competition on intrinsic motivation in exergames and the conditional indirect effects of presence, *in* 'annual conference of the International Communication Association, Singapore'.

Thomas, S., Reading, J. and Shephard, R. J. (1992), 'Revision of the physical activity readiness questionnaire (par-q).', *Canadian journal of sport sciences* .

Warburton, D. E., Bredin, S. S., Horita, L. T., Zbogar, D., Scott, J. M., Esch, B. T. and Rhodes, R. E. (2007), 'The health benefits of interactive video game exercise', *Applied Physiology, Nutrition, and Metabolism* **32**(4), 655–663.

Yim, J. and Graham, T. (2007), Using games to increase exercise motivation, *in* 'Proceedings of the 2007 conference on Future Play', ACM, pp. 166–173.

Appendix A

University Health Screening Questionnaire

HEALTH SCREEN **Participant Number** ....

It is important that volunteers participating in research studies are currently in good health to exercise. This is to ensure (i) their own continued well-being and (ii) to avoid the possibility of introducing bias into the study outcomes.
Please complete this brief questionnaire to confirm your eligibility to participate:

Further Health Screen continued overleaf…

1. **At present,** do you have any health problem for which you are:
   (a).. on medication, prescribed or otherwise........................ Yes No

   (b).. attending your general practitioner.............................. Yes No

2. As far as you are aware, **do you suffer** or **have you ever** suffered from:
   (a).. Convulsions/epilepsy.. Yes No

   (b).. Asthma... Yes No

   (c).. Pressure sores... Yes No

   (d).. Diabetes.. Yes No

   (e).. A blood disorder.. Yes No

   (f).. Head injury.. Yes No

   (g).. Digestive problems... Yes No

   (h).. Heart problems... Yes No

   (i)... Disturbance of balance/coordination............................ Yes No

   (j)... Disturbance of vision.. Yes No

   (m). Ear / hearing problems.. Yes No

   (n).. Thyroid problems.. Yes No

   (o).. Kidney or liver problems... Yes No

   (p).. Urinary tract infection... Yes No

   (q).. Cognitive impairment.. Yes No (s).. *Autonomic dysreflexia .... Yes No

4. ***Has any, otherwise healthy, member of their family under the age of 35 died suddenly during or soon after exercise**

...... Yes No

a) If YES to any question, please describe briefly if you wish (eg to confirm problem was/is short-lived, insignificant or well controlled.)

…..……

……………………………………………………………………………………………………..

…..……

……………………………………………………………………………………………………..

…..……

……………………………………………………………………………………………………..

…..……

……………………………………………………………………………………………………..

b) Questions indicated by (*) requires your Doctor to fill out the 'Doctors Consent Form provided'

Signature:____.Date: .

Thank you for your cooperation!

Doctors Consent Form

Doctor's Consent for Exercise Testing and Advice Form

I (insert name)

am planning to participate in a supervised series of sports performance tests having first completed an informed consent and medical questionnaire. A participant information sheet describing these tests is attached. The tests undertaken may include exercising to exhaustion, (subject to your recommendation). Please would you confirm that I am medically fit to complete the tests under the supervision of sports scientists.

Doctor's Statement

I (insert name)
following examination of your medical details, consider you **SAFE/ NOT SAFE** to proceed, subject to my comments below.

Comments:

Practice Address:-

Signed : **Date :**

Appendix B

Participant Instruction Sheet

Participant Instruction Sheet

Welcome, and thanks for participating in this study. It is important that you make an informed decision about whether or not to take part, so we ask that you take some time to read this participant information sheet carefully and discuss it with others if you wish. You will then have the opportunity to ask any questions you may have before deciding whether to volunteer and will of course be free to withdraw from the study at any time without giving a reason. Today you will be exercising through three short High Intensity Interval Training (HIIT) sessions while playing a virtual reality (VR) game, as well as filling out questionnaires based on your experience. The total session is expected to take no longer than 60 minutes.

Your participation will help us understand how VR exergames can help people to have an enjoyable and effective exercise experience. All data collected during this study are anonymous and are only used for the purpose of measuring and improving VR exergames. If you are unwilling to be filmed during please alert the study conductor.You can withdraw all the data collected from you within two weeks after data collection by emailing the experimenter (see email below). All information will be subject to the current conditions of the *Data Protection Act 1998* to protect your anonymity.

Setup

Before you begin the game you will first need to read through this sheet and sign a consent form to confirm that you understand the procedure and your rights. Next, you will fill out the PAR-Q questionnaire again to determine whether it is safe for you to exercise today.

If you are not feeling well or at any point during the study need a break, please let the experimenter know immediately. You can stop the study at any time without having to give a reason.

You will be asked to fill out a pre-experiment questionnaire about your demographics, frequency of exercise and attitude towards exercise.

After completing these, you will be given privacy to change into your appropriate sportswear if necessary then instructed on how fit the heart rate monitor to your chest. Measurements of height, weight and body composition will then be taken and recorded.

Following this the study conductor will help you calibrate the VR headset so that the in-game view corresponds to your comfortable sprint position on the bike. You will be asked to put on the headset, take your sprint position and the study conductor will commence the setup. It is important that you are in a comfortable sprinting position as you should try to maintain this posture throughout the experiment.

When assuming this position ensure you have a sturdy hold on the exercise bike as it is compulsory you are in full control at all times.

You will also be shown the Borg RPE scale which you will use after the sprint sections to rate your perceived exertion.

The Game

You will undergo 3 separate High Intensity Interval Training sessions while you play a simple “racing” esque game. The goal is to avoid the trucks as these will slow you down, and beat your previous recordings (in sessions 2 and 3).

Each session is broken down into **5 phases**:

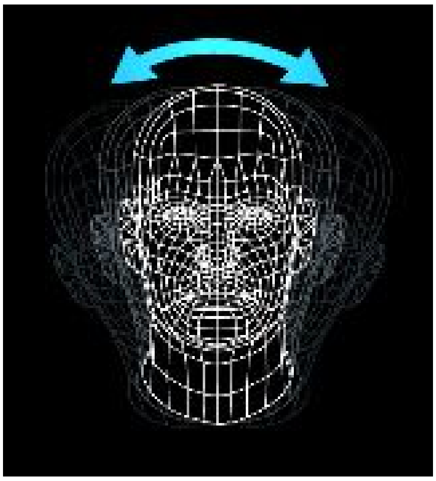
1. A 60 seconds warmup.
2. A 30 second sprint.
3. A 90 second recovery.
4. A 30 second sprint.
5. A 90 second recovery/cooldown.

In the low intensity sections (warmups, recoveries and cooldowns) the distance you travel does not matter but you should try to maintain **65 - 70 rpm**. The low intensity sections are in the daytime as seen in the picture below.

In the high intensity sections you should be sprinting on the bike as fast as you can (however only work as hard as you normally would in a regular gym environment), in the first session you will simply record your performance but in sessions 2 and 3 your goal in the high intensity sections is to **beat** your "ghost". The ghost looks just like your character. The high intensity sections are at night and you will be being chased by the police, as seen below.

The Controls

The game is in virtual reality and your body is the controller. You will **cycle** faster on the bike to move faster forward in the game. You will **lean/tilt** your head left to move left across the track and similarly right to move right across the track. See the diagram below.



You can otherwise **turn** your head and tilt/lean up and down to look around at your environment as you play.

The Display Interface

When using the Virtual Reality Headset you will see the below display.

- The number marked **A** is the **time left in the current phase**.
- The number marked **B** will not appear in your first session but will in session 2 and 3. This number is the **distance you are ahead or behind your ghost** if the number was -5m you are 5m behind your ghost.
- The number marked **C** displays your current **bike RPM**.
- The text marked **D** will change to give you **prompts** throughout as to what you should be doing i.e. getting ready to sprint, getting ready to slow down, etc.

Training Session

Once the system is calibrated you will partake in a short training session to ensure you understand the game's format, allow you to get familiar with the controls and the bike, and to ensure that you are comfortable doing so.

If during this training session you have any concerns or questions please do ask.

Sessions and Questionnaires

There are 3 sessions as previously mentioned during each one after the sprints you will be asked to rate your perceived exertion on the Borg RPE scale.

- You will complete the first HIIT session thus recording your “ghost”.
- You will then fill out a short questionnaire, and be allowed to drink some water.
- You will then complete the second HIIT session - racing your ghost in the sprint sections.
- You will then similarly fill out another short questionnaire and be allowed to drink some water.
- You will the complete the third HIIT session - racing your best ghost in the sprint sections.
- You will complete another post session questionnaire and be allowed more water.

Post Experiment

After the final session you will be asked to fill out one final questionnaire and then will be subsequently debriefed on the experiment.

Risk

Playing the exergame involves cycling on a stationary exercycle at an intensity ranging from very light (about 50% of maximum exertion) to very hard (about 90% of maximum exertion). The exercise will cause your heart rate to rise and you will likely feel a shortness of breath, start sweating and feel physically exhausted. You may also experience muscle soreness during and after the exercise.

One of the keys to safety in HIIT training is to change the intensity of the sprint to suit your individual preferred level of challenge. **You should carefully choose your own optimal sprint intensity. Safety in participation should always be the primary priority.**

There are some risks with physical activity(HIIT or any exercise training) as it involves a degree of physical exertion.Persons who have been living rather sedentary lifestyles or periods of physical inactivity may have an increased coronary disease risk to high intensity exercise. Family history, cigarette smoking, hypertension, diabetes (or pre-diabetes), abnormal cholesterol levels and obesity will increase this risk.

We mitigate these risks by making sure that all exercise will be performed under supervision in a safe environment, with participants that can be considered low risk. All participants will fill out a health screen questionnaire prior to participating in the study and if necessary, will need written consent giving clearance to exercise from a GP. Musculoskeletal injury and complications will be reduced by warm up and cool down periods, stretching and gradual progression of exercise intensity

It is not uncommon for people to get motion sick using virtual reality headsets. We have done our best through play-testing and adhering to the findings of studies in the field, to ensure there are no sensory disconnects to encourage this. That being said the effect this can have must be handled on a case to case basis. If you experience any motion sickness or discomfort in the training session please alert the study conductor.

If at any point during the study you start to feel nauseous or motion sick, please alert the study conductor as it as paramount to your safety not to continue in this state of wellbeing.

Participant Rights

You have the right to withdraw from this study at any time without giving reason. You can withdraw all the data collected from you within two weeks after data collection by emailing the study conductor (see email below). Additionally, you have the right to not answer any of the questions in the post-game questionnaire. If you have any questions about the study then please ask them now, or email the experimenter Alex Whaley at a later date at aw709@bath.ac.uk

Appendix C

Participant Consent Form

Participant Consent Form

☐ The nature, aims and risks of the research have been explained to me

☐ I have read and understood the Participant Instruction Sheet.

☐ I understand what is expected of me

☐ I understand that I can withdraw from the study at any time without giving a reason

☐ Any questions I have about my participation in this study have been answered satisfactorily.

☐ I am taking part in this study voluntarily.

☐ I consent to the processing of my personal information for the purpose of this research study

☐ I understand that my information will be treated as strictly confidential in accordance with the Data Protection Act 1998

☐ I understand that this study involves high-intensity exercise which may lead to breathlessness, discomfort and an increased coronary disease risk.

☐ I am willing to be video recorded whilst taking part in this

☐ I understand that my identity will not be linked to any data retrieved in this study.

☐ I am willing to be video recorded whilst taking part in this exercise study.

☐ I am confident in my ability to safely cycle nine 30 second sprints, in a single 30 minutes session, on an exercise bike.

☐ I feel well enough to exercise today.

☐ I have not eaten in the past 2 hours.

☐ I am aware of the potential risks of using virtual reality headgear, whilst physically exerting myself.

☐ I understand that I am assigning my rights to the data collected during the study to the University of Bath.

☐ I agree not to talk about the design of the experiment to others until data from all participants has been collected.

☐ I agree to volunteer as a subject for the study described in the Information sheet and I give full consent to my participation in this study

☐ [OPTIONAL] I would like to hear about the results of this study.

I, the participant, consent to all points made above:

Signature: **Date:**

Bath Email:

I, the study conductor, promise to protect the confidentiality of the participant, and will only use the data collected in this study for research purposes:

Signature: **Date:**

Appendix D

Ethics Checklist

13-POINT ETHICS CHECKLIST

Have you prepared a briefing script for volunteers?
A full participation instruction sheet detailing the entire structure of the experiment is given to and must be read completely before participants are allowed to take part in the study. The sheet details the exact procedure, what will be physically and mentally required of them, as well as informing them of any potential risks/effects of the study.

Participants data is anonymous and the kind of data collected is detailed in the instruction sheet, however they must also sign a consent form consenting to the collection of said specified data and it's ownership by the University of Bath. This consent form also confirms they have fully read the instruction sheet and that all their questions have been answered.

Will the participants be using any non-standard hardware?
The focus of the experiment involves the use of an exercise bike as well as an HTC Vive virtual reality headset. In the consent form users must confirm they are capable of the intended exercise machine using an exercise bike.

The potential risks of using a virtual reality headset whilst physically exercising are mitigated as much as possible through the game and experiment design - most importantly virtual reality induced motion sickness. A training period will be carried out prior to the session to ensure the participant is comfortable using the setup and experiences no physical or mental discomfort. If this changes during the study students maintain the right to withdraw at any time.

Is there any intentional deception of the participants?
The studies focus is investigating whether users can improve at a faster rate by increasing resistance without informing the user. This increase is very marginal and making sure we determine an initial resistance that users are comfortable performing high intensity exercise at is crucial to ensuring this manipulation doesn't negatively affect users.

We will also by monitoring through questionnaires exactly how the user experience and user satisfaction changes before/after this factor is introduced.

How will participants voluntarily give consent?
A full signature required consent form is included (See Appendix), as well as a full participation instruction sheet detailing the details of the study (See Appendix) which is compulsory to read.

Will the participants be exposed to any risks greater than those encountered in their normal work life?
The only significant potential risk that can be incurred when using virtual reality is that of motion sickness. This risk is increased by intense exercise, however the user is sat down and has a good grip on the bike to stay stable. This is discussed further in the experimental design.
Also the game was designed to reduce the sensory disconnect between user actions and in game actions as much as possible. Certain game features had to be removed to ensure the study was as safe as possible. We conducted extensive playtesting to ensure users felt safe whilst using equipment.

Are you offering any incentive to the participants?

No users are not being offered any incentive since the performance we are investigating is relative, we don't want to skew user improvement by adding any incentive to improve other than the game elements we have incorporated.

Are any of your participants under the age of 16?
No we will only be using people over the age of 16, primarily university students and other people from the computer science department.

Do any of your participants have an impairment that will limit their understanding or communication?
No, none of our participants fell into this category.

Are you in a position of authority or influence over any of your participants?
No, all of our participants volunteered of their own free will, and are required to confirm this in the consent form they sign prior to the study.

Will the participants be informed that they could withdraw at any time?
The participant instruction sheet clearly specifies that it is within participants rights to withdraw at any time, participants are explicitly told if they ever start to experience any discomfort they should withdraw.

Will the participants be informed of your contact details?
All participants will be given the full personal details of myself the study conductor as well as supervisor/ member of staff Christof Lutteroth, this will be given to them on a sheet of paper during the debrief.

Will participants be debriefed?
After the experiment participants will be told the exact nature of the experiment they just took part in, the debrief will clarify on exactly how we will make use of their data, and the full nature of the paper I am writing. Participants will also be asked to not talk about the nature of the experiment with any other participants prior to the full set of data being collected.

Will the data collected from the participants be stored in an anonymous form?
Participants are assigned ID's with no reference to their identification, we can use ID's to reference specific pieces of data without having any reference to exactly what participant it belonged to.

Appendix E

Questionnaires

Due to the number of questionnaires we used and their length, they could not be included in the physical dissertation appendix as there would be be hundreds of pages of appendixes.

The Questionnaires are all stored in the questionnaire folder on the provided DVD.

Appendix F

Raw Data

Due to the size of the results tables and the number of questionnaires we used, the raw data could not be included in the physical dissertation appendix as there would be be hundreds of pages of appendixes.

The Raw Data .csv's are all stored in the raw data folder on the provided DVD.